“智慧政务”中的文本挖掘应用

摘要

随着大数据、 云计算、人工智能等技术的发展，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，对提升政府的管理水平和施政效率具有极大的推动作用。

问题一建立一级标签分类模型，首先对数据文本分词，利用 Keras 分词器 Tokenizer 生成编码字典。将标签序列统一转为 one-hot 编码，使用 softmax 激活函数构建多分类 TextCNN 卷积神经网络模型学习文本特征。最后根据附件 2 的留言内容数据，随机选取 75%进行模型训练，25%来进行模型测试，并采用 F-Score 系数进行模型评价。

问题二，首先对数据进行预处理包括数据清洗，jieba 文本分词处理，词性标注并获取指定词性的词语，停用词处理，消歧词典进行消除歧义处理，删除无用单字。其次使用 Countvectorizer 接口对文本进行特征提取，输出稀疏矩阵，对稀疏矩阵使用 DBSCAN 聚类，使用聚类得各类的数据量作为热度评价指标，将离群点标记为-1，将各类依数据量排名，数据量越多对应热度排名越靠前。进一步分析处理并汇总整理出热带问题表和热点问题留言明细表。最后使用 TextRank 算法对热点话题排行，并提取对应留言内容摘要作为上述排名的进一步检验。

问题三，先利用 TextCNN 分类模型对附件 4“留言详情”进行一级标签的分类，并将标签转换为文字表示后对应合并到附件 4。定义答复质量评价标准，并对答复意见进行人工标注。然后利用 TF-IDF 算法对答复意见进行关键词提取，提取出来的各类别关键词保存为 txt 文件，作为后续操作的关键词词典。以交通运输这一类别（311 条答复意见）为例，通过关键词库去筛选文本，使其仅保留该类别的关键词，将文本转换为数值向量，作为卷积神经网络（CNN）的输入。并按 3:1 的比例拆分成训练集与测试集，答复质量的评价结果作为神经网络的输出。该模型共两次使用卷积神经网络（CNN），可以对留言答复分类并对留言答复进行质量评价。

关键词：TextCNN, DBSCAN 聚类, 热点挖掘, 关键词提取, TF-IDF

目录

1 绪论

1.1 背景、目的及意义

近年来，随着微信、微博、市长信箱、阳光热线等网络问政平台逐步成为政府了解民意、 汇聚民智、凝聚民气的重要渠道，各类社情民意相关的文本数据量不断攀升，给以往主要依靠人工来进行留言划分和热点整理的相关部门的工作带来了极大挑战。同时，随着大数据、云计算、人工智能等技术的发展，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，对提升政府的管理水平和施政效率具有极大的推动作用。

本次数据挖掘的数据来自互联网公开来源的群众问政留言记录，及相关部门对部分群众留言的答复意见。我们将用自然语言处理和文本挖掘的方法找出文本中蕴含的有价值的信息。

1.2 相关工作

自然语言处理是机器学习的研究热点。在当前的文本分类研究中，机器学习方法大量使用了基于统计学的方法模型。最早的用于文本分类传统模型是朴素贝叶斯模型(NB) ，其次 K 近邻，SVM，神经网络，决策树等监督学习方法也都应用在了文本分类领域，并取得了高性能的分类效果。

相较于国外，国内对话题热点挖掘的研究相对较晚，英文可直接通过空格进行分词，而中文则需要考虑词性语义等多种因素，因此需要结合语义的机器学习方法进行分词，但仍然是取得了很好的成果。王翔使用了基于句义结构模型构建了一种基于聚类的互联网热点事件发现方法，采用 single-pass 聚类思想和凝聚式层次聚类与 K-Means 聚类算法相结合的聚类算法来发现新闻热点[1]。王馨在新闻挖掘过程中使用了改进的关联规则算法，根据交互信息来计算文本字符串的相似度，然后得出热点新闻的关键词集合，再进行热度计算来研究新闻热点[2]。陈龙引入了 LDA 主

题模型，提出了一种多核心话题描述模型，能够识别同一话题下不同的关注核心，之后采用划分聚类与层次聚类结合的方法对新闻报道进行精确聚类[3]。

本文的主要工作是对留言进行文本分类、对留言进行热点挖掘及设立答复评价模型。

2 问题分析

2.1 问题一的分析

本题需要根据附件 2 中的留言内容（留言主题和留言详情）对留言进行一级标签分类，因此需要对留言文本进行分词，进而建立文本卷积神经网络(TextCNN)分类模型训练文本得到分类预测标签，与原标签比对，最后结合 F-Score 对文本的分类方法进行评价。

2.2 问题二的分析

问题二需要我们对热点问题进行挖掘，热点问题即“某一时段群众集中反映的某一问题”。我们需要根据附件 3 的数据，分析某一时间段内的对于特定地点或特定人群的问题归类，同时定义科学合理的热度评价指标，并给出评价结果，生成排名前五的热点问题信息按照热点问题表、热点问题留言明细表格式给出汇总结果，大致思路为：

（1）数据预处理：数据清洗，jieba 文本分词处理，词性标注并获取指定词性的词语，停用词处理，消歧词典进行消除歧义处理，删除无用单字。

（2）使用 Countvectorizer 接口对文本进行特征提取，输出稀疏矩阵

（3）对稀疏矩阵使用 DBSCAN 聚类

（4）使用聚类得各类的数据量作为热度排名依据，将离群点标记为-1，将各类依数据量排名，数据量越多对应排名越靠前

（4）TextRank 算法对热点话题排行，并提取对应留言内容摘要作为上述排名的进一步检验

在后续分析中，将对实际操作对应上述步骤进一步讨论。

2.3 问题三的分析

针对问题三，先按照问题一中建立的卷积神经网络(CNN)分类模型对附件 4 根据留言详情这一列进行一级标签的分类，并将所得分类后的标签转换为文字后一一对应合并到附件 4，并按定义的评价标准人工标注答复质量。然后利用 TF-IDF 对各类别的答复意见进行特征提取，提取出来的各类别关键词保存为 txt 文件作为后续操作的关键词词典。以交通运输这一类别（311 条答复意见）为例，通过关键词库去筛选文本，使其仅保留该类别的关键（特征）词，按 1:3 拆分成训练集与测试集。然后再次应用到卷积神经网络(CNN)，答复质量的评价结果作为神经网络的输出。该模型可以对留言答复分类并对留言答复进行质量评价。

3 数据预处理

3.1 问题一&问题三

基于附件 2 中的留言内容，为使得留言文本可放入 CNN 模型进行训练，我们需要做以下预处理：

首先，利用 jieba.cut 对文本进行分词；其次，按分词后得到的词语用 Keras 分词器 Tokenizer 生成编码字典，并且字典按词频从高到低保留数量；再将文本转序列，用于神经网络的 Embedding 层输入(Tokenizer.texts_to_sequences)；再次，获取标签集合，用于 one-hot；最后，构建标签 one-hot，遍历标签列表生成 one-hot 二维数组。

3.2 问题二

附件 3 包括留言编号、留言用户、留言主题、留言时间、留言详情、反对数和点赞数 7 项指标。

在问题二的数据预处理中，我们首先定义时间函数归类不同时间段的留言数据，再从清理未知字符、清理空白字符、将英文大写转小写和清理超链接等方面对留言主题和留言内容清理。

其次，使用 python 的 jieba 库对留言数据进行分词处理，将字符串分割输出若干词汇构成的词向量。定义 get_words_by_flags()函数对该向量获取指定词性的词语，本研究中默认提取名词和动词作为热点问题信息的特征要素。

此外，考虑到停用词本身为没有太大意义的词语例如助词和语气词，实验中对应停用词词典将分词中的停用词删除；对于词义消歧操作，选用基于词典的神经中文词义消歧。该工作的思想是计算语义词典中各个词义的定义与上下文之间的覆盖度，选择覆盖度最大的作为待消解词在其上下文下的正确词义。最后识别并删除词向量中的无用单字。

4 基于 TextCNN 的留言多分类

卷积神经网络（Convolutional Neural Networks, CNN）是一类包含卷积计算且具有深度结构的前馈神经网络，是深度学习（Deep learning）的代表算法之一。卷积神经网络仿造生物的视知觉机制构建，可以进行监督学习和非监督学习，它的人工神经元可以相应一部分覆盖范围内的周围单元，对处理大型图像表现出色，同样也可以运用在文本分类[4]。

4.1 TextCNN 神经网络结构

政府平台的民众留言反应了多方面的民生问题，因此需要进行文本多分类，而 CNN 神经网络结构可以较好的解决多分类问题。留言分类的训练模型由嵌入层、卷积层、池化层和全连接层所构成。卷积神经网络的底层是由嵌入层，卷积层和池化层组成，在保持特征不变的情况下减少维度空间和计算时间；更高层次是全连接层，其输入是由卷积层和池化层提取得到的特征；最后一层是全连接层，可以是一个分类器，若目标为分类，采用 softmax 激活函数进行多分类；若为二分类，采用 sigmoid 激活函数进行分类[6]。

4.1.1 嵌入层(Embedding Layer)

通过一个隐藏层，将 one-hot 编码的词投影到一个低维空间中，本质上是特征提取器，在指定维度中编码语义特征。因此，语义相近的词，它们的欧氏距离或余弦距离也比较近。

4.1.2 卷积层(Convolution Layer)

在处理图像数据时，CNN 使用的卷积核的宽度和高度的一样的，但是在 TextCNN 中，卷积核的宽度与词向量的维度一致。由于我们的输入是一个句子，句子中相邻的词之间关联性很高，因此，当我们用卷积核进行卷积时，不仅考虑了词义，而且考虑了词序及其上下文。

4.1.3 池化层(Pooling Layer)

在卷积过程中，我们使用了高度为 3 的卷积核，卷积层后得到的向量维度会不一致，所以在池化层中，我们使用 1-Max-pooling 将每个特征向量池化成一个值，即抽取每个特征向量的最大值表示该特征，而且认为这个最大值表示的是最重要的特征。当我们对所有特征向量进行 1-Max-Pooling 之后，还需要将每个值拼接起来。得到池化层最终的特征向量。在池化层到全连接层之间可以加上 dropout 防止过拟合。

4.1.4 全连接层(Fully Connected Layer)

假设有两层全连接层，第一层用‘relu’作为激活函数，第二层则使用 sigmoid 激活函数得到属于每个类的概率。

4.2 模型训练

由于附件 2 中的留言文本已有分类标签，因此我们采用监督学习对样本进行卷积神经网络的训练学习。

我们经过预处理将数据转化为模型的输入矩阵，利用 Tokenizer 生成词向量，模型训练相关参数定义如下：训练时期（epochs）为 10，批量大小（batch_size）为 64，词向量（num_words）大小为 3000，每个样本设置为定长为 85 的序列，卷积核个数为 128，同时卷积层采用 Relu 激活函数[5]。生成的模型如下图 4-1 所示：

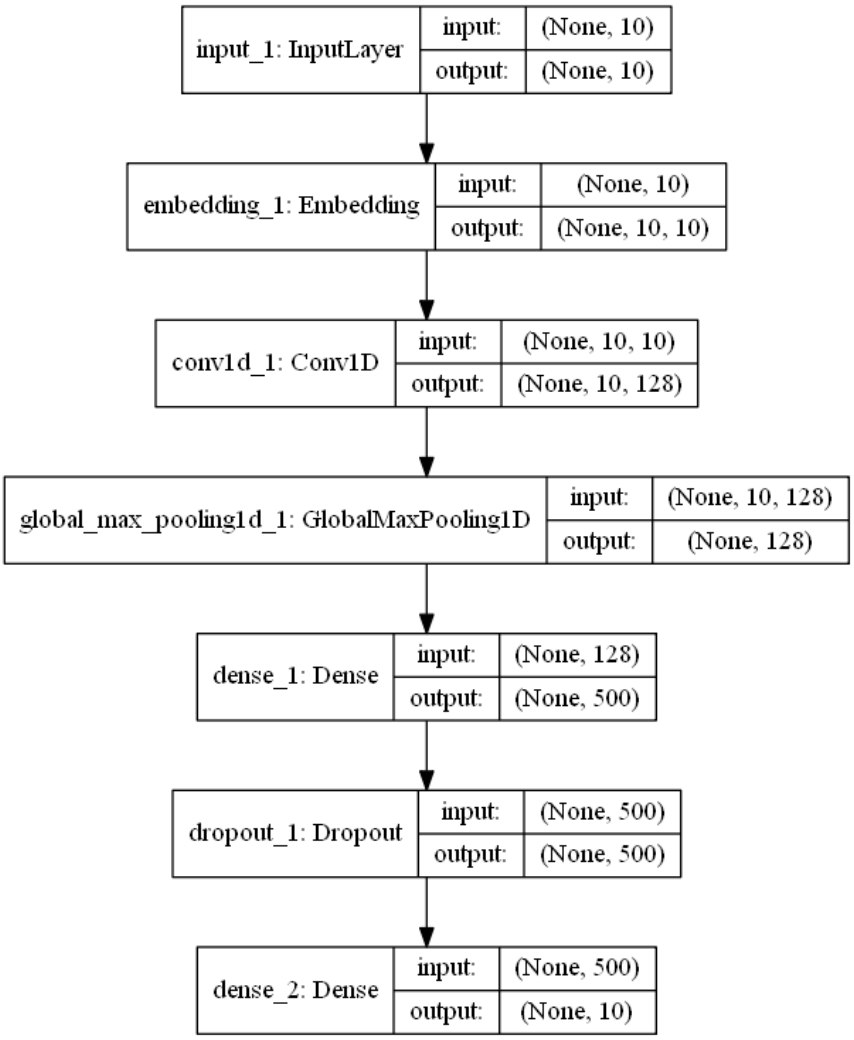


图 4-1 训练模型

使用 train_test_split 拆分训练集和测试集，训练集取 75%，测试集取 25%。经过 TextCNN 学习不同类别文本的特征，输出 one-hot 序列的标签集，最后与原数据标签进行比对，得到训练集正确率和损失率如下图 4-2 所示：

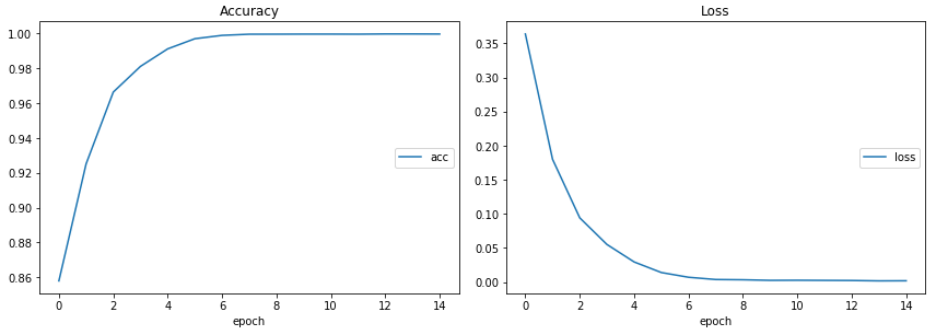


图 4-2 训练集正确率和损失率

上图说明训练集的正确率与损失率呈负相关，且训练正确率逐渐趋于 100%，损失率趋于 0.2%。

4.3 训练结果

F1-Score 可以通过 sklearn.metrics 中的 classification_report 得出，用于评价选用分类方法的优劣，公式如下：

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

其中 P_i 为第 i 类的查准率， R_i 为第 i 类的查全率。

调整参数后所得结果如下表 4-1：

表 4-1 F1-Score 评价

Epochs	Batch_size	F1-Score
10	64	0.8705
20	64	0.8534
30	64	0.8494
10	128	0.8622
20	128	0.8652
30	128	0.8646

由表 4-1 可得，当 epochs=10 , batch_size= 64 时 F1-Score 得分最高，为 0.8705，此外，一级分类各个类别的得分如下表 4-2 所示：

表 4-2 一级分类 F1-Score

	劳动和社会保障	教育文体	商贸旅游	交通运输	卫生计生	城乡建设	环境保护
F1-Score	0.9317	0.7625	0.8682	0.9332	0.8458	0.9017	0.8503

综合以上两表格，说明 TextCNN 模型对留言文本分类的效果较好，但并非十分理想。

4.4 模型改进

为提高文本分类的准确性，TextCNN 模型的改进方向[6]如下：

- 尝试使用高度为 4, 5 的卷积进行半池化。
- 循环叠加卷积与半池化：通过增加网络的深度，获得单词与单词之间的特征关系，增加特征维度。

5 基于 DBSCAN 聚类的热点挖掘[8]

5.1 算法思想及流程

5.1.1 基于 CountVectorizer 文本特征提取

在数据预处理后，定义函数对留言样本数据集合获取词库、单频词列表、最常见的词构成列表和列表中不重复的值的个数以便于特征选择等分析操作。

对文本进行特征提取，使用的接口 CountVectorizer 主要考虑词汇在文本中出现的频率，它在单个类中实现标记化和计数，对每一文本输出关于词语的稀疏表示再得到总的稀疏矩阵。

5.1.2 DBSCAN 聚类

我们进一步使用 DBSCAN 聚类算法[7]对该稀疏矩阵处理，中文含义为“基于密度的带有噪声的空间聚类”。该算法将具有足够密度的区域划分为簇，并在具有

噪声的空间数据库中发现任意形状的簇，它将“簇”定义为密度相连的点的最大集合。

DBSCAN 的簇里面可以有一个或者多个核心对象。如果只有一个核心对象，则簇里其他的非核心对象样本都在这个核心对象的 ϵ-邻域里；如果有多个核心对象，则簇里的任意一个核心对象的 ϵ-邻域中一定有一个其他的核心对象，否则这两个核心对象无法密度可达。这些核心对象的 ϵ-邻域里所有的样本的集合组成的一个 DBSCAN 聚类簇。

如何找到这样的簇样本集合呢？DBSCAN 使用这样的方法：它任意选择一个没有类别的核心对象作为种子，然后找到所有这个核心对象能够密度可达的样本集合，即为一个聚类簇。接着继续选择另一个没有类别的核心对象去寻找密度可达的样本集合，这样就得到另一个聚类簇，直到所有核心对象都有类别为止，如图 5-1 所示。

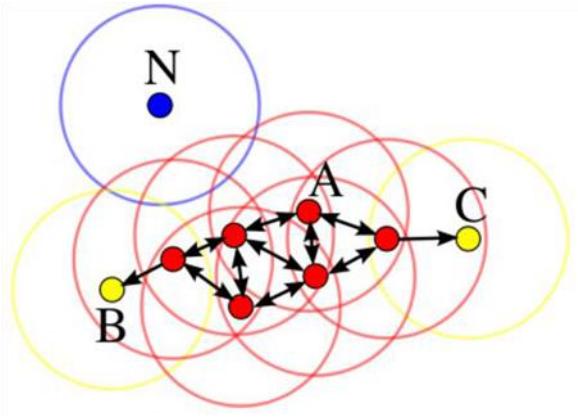


图 5-1 其中 A 为核心点，B 和 C 表示边界点以及 N 表示离群点

使用聚类算法中最常用的评估方法—轮廓系数（见下式），对聚类结果评估。

$$\mathrm{s(i)} = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

1. 计算样本i到同簇其它样本到平均距离$a(i)$。$a(i)$越小，说明样本i越应该被聚类到该簇（将$a(i)$称为样本i到簇内不相似度）

2. 计算样本i到其它某簇C_j的所有样本的平均距离$b(i)$，称为样本i与簇C_j的不相似度。定义为样本i的簇间的不相似度为：$b(i)=\min(\mathrm{bi}_1, \mathrm{bi}_2, \ldots, \mathrm{bi}_k)$

计算得到$\mathrm{s}(i)$，且值接近 1，说明样本i聚类合理。

DBSCAN 算法相比 K-means 算法具有以下优点：

- 原始数据分布规律没有明显要求，能适应任意数据集分布形状的空间聚类，因此数据集适用性更广，尤其是对非凸装、圆环形等异性簇分布的识别较好。
- 无需指定聚类数量，对结果的先验要求不高
- 由于 DBSCAN 可区分核心对象、边界点和噪点，因此对噪声的过滤效果好，能有效应对数据噪点。

DBSCAN 算法对整个数据集进行操作并且聚类时使用了一个全局性的表征密度的参数，经过若干试验，最终使用的参数如下表 5-1。

表 5-1 参数选择

参数	eps	min_samples	metric
值	0.4	4	cosine

对于聚类得到的具有标签的数据，将每一标签值包含的数据点数量转化为排行，该类数据数量越多则排名越靠前，而离散点标记排名值为-1。对留言主题数据进行聚类处理得到 68 类（见 df_title_rank.csv）。同样对留言内容数据，聚类处理得到 31 类（见 df_content_rank.csv）。整合后得到热点问题表和热点问题留言明细表。

5.1.3 TextRank 算法验证

此外，我们使用 TextRank 算法[9]获取文本的摘要对应热点主题展示并用于热点问题的检验，TextRank 算法是一种用于文本的基于图的排序算法，其基本思想来源于谷歌的 PageRank 算法, 通过把文本分割成若干组成单元(单词、句子)并建立图模型, 利用投票机制对文本中的重要成分进行排序, 仅利用单篇文档本身的信息即可实现关键词提取、文摘，具有简洁有效的特点。

5.2 实验结果及分析

5.2.1 DBSCAN 聚类结果说明

对留言主题数据进行聚类处理得到 68 类（见 df_title_rank.csv）。同样对留言内容数据，聚类处理得到 31 类（见 df_content_rank.csv）。

在原先基础上，留言详情中附加了词性识别，而 title_cut 和 content_cut 为留言主题和留言内容经过数据预处理后的分词结果，title_rank 和 content_rank 为依聚类各类数量定义的热度排名，排名越靠前即热度越大。

整理汇总各类数据并提取对应时间段、地点与对象整合成相应格式的结果。

5.2.2 聚类后热度排名评价说明

对聚类得到的对应类别 label 的数据，保留 label 值为-1 的数据排名为-1，标记为离群点。其他依据该类中的数据量排名，对各类进行排名。

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	54	2019/7/2 至 2019/12/31	A 市伊景园滨河苑	伊景园滨河苑捆绑销售车位
2	2	45	2019/11/2 至 2020/1/26	A 市 A2 区丽发新城小区	违建搅拌站噪音灰尘扰民
3	3	17	2019/7/21 至 2019/12/4	A4 区 A7 县君悦幼儿园	魅力之城小区一楼夜宵摊严重污染空气
4	4	8	2019/3/13 至 2019/12/23	A7 县诺亚山林小区	反对小区门口设立医院
5	5	7	2019/6/10 至 2019/9/2	A7 县星沙四区凉塘路	旧城改造拖到何时动工

上表为整合汇总得到的热点问题表，其中热度指数表示该问题 ID 对应的留言总数。

在聚类后的整合汇总工作中，我们进一步对该类别对应的留言内容通过文本模糊查询与筛选、相似度计算增添未被聚类算法包含的同义的留言内容。以排名为 1 的类别为例，聚类得到排名为 1 的数据（见表 df_content_rank.csv），此时有一部分经过人工识别地名对象，发现并不应该归于该类。通过进一步查询“丽发新城”和“搅拌”以及文本相似度计算，整合更多留言内容详情汇总为热点问题留言明细表。

5.2.3 TextRank 算法对热点话题及摘要结果展示

结果如下图 5-2 图 5-2 所示：

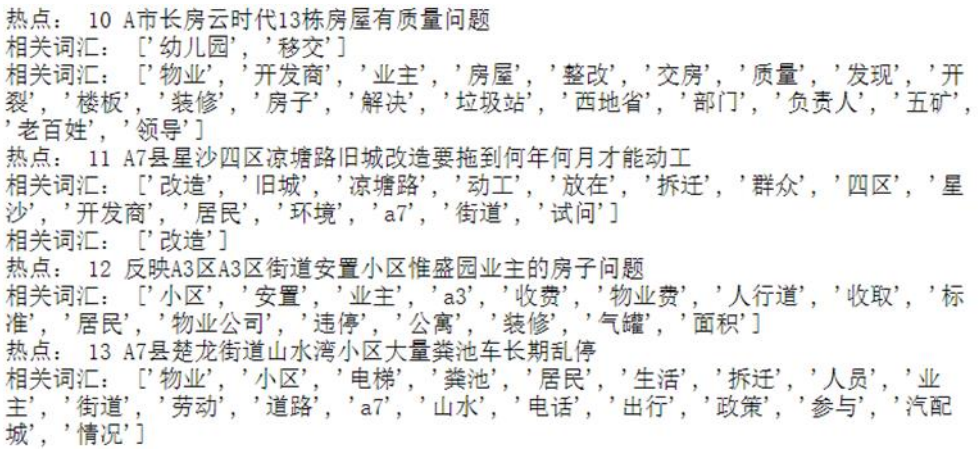


图 5-2 TextRank 算法提取热点话题

依据主题热度排名，输出提取的各主题对应的留言内容摘要，以此作为进一步检验。

5.2.4 分析与改进

（1） 在结果分析 2 中提到的聚类结果为 1 和 2 的经过进一步筛选与相似度计算分析对收集的留言详情改进，增加该类对应的留言内容，减少该类中原本错误归于该类的留言内容。同样检验类别 3、4、5，对于这些类可以看到聚类的分类效果较好。

（2） 本工作中选用聚类算法中的 DBSCAN 算法，其具有以下弱点：

- 对于高纬度问题，基于半径和密度的定义成问题。
- 当簇的密度变化太大时，聚类结果较差。
- 当数据量增大时，要求较大的内存支持，I/O 消耗也很大。

可以尝试其他有效的聚类算法进行本问题的分析。

6 答复意见评价方案设计

6.1 方案设计思想

传统的答复质量评价基于文本的结构化特征，例如，文本长度，标点符号占比，平均句子长度，成词率等，这种方法能够一定程度反应答复质量。但是这种情况未考虑答复风格对于上述特征的影响，两个文风不同但质量相同的答案会出现明显的结构化差异，有学者[10]曾经分析过金庸与古龙小说的文本结构化特征，结果显示造成结构化特征不一致的原因是小说作者的写作风格不同。

而本方案的提出是认为答复质量取决于其所包含的语义信息，语义信息是信息的表现形式之一，指能够消除事物不确定性的有一定意义的信息[11]。一个答复中包含的语义信息越多，那么该答复的质量越高。而一个答复即使篇幅很长，涵盖的有用信息很少，那么该答复也不能算是一个优质答复。比如以下这个答复，文本长度是别的答复近十倍，是因为列举多个文件规定的具体信息，但其中包含的对问题反映者有用的信息只有寥寥几句。而且通过语义信息可以体现出答复意见与留言所提问题的相关程度及对问题回答的完整性。

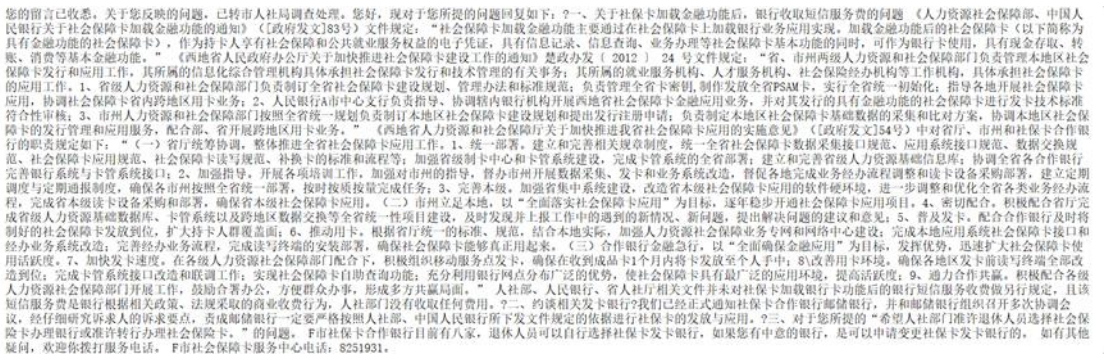

您的留言已收悉，关于您反映的问题，已转市人社局调查处理。您好，现对于您所提的问题回复如下：？一、关于社保卡加载金融功能后，银行收取短信服务费的问题 《人力资源社会保障部、中国人民银行关于社会保障卡加载金融功能的通知》（《政府发文》83号）文件规定：“社会保障卡加载金融功能主要通过在社会保障卡上加载银行业务应用实现。加载金融功能后的社会保障卡（以下简称为具有金融功能的社会保障卡），作为持卡人享有社会保障和公共就业服务权益的电子凭证，具有信息记录、信息查询、业务办理等社会保障卡基本功能的同时，可作为银行卡使用，具有现金存取、转账、消费等基本金融功能。” 《西地省人民政府办公厅关于加快推进社会保障卡建设工作的通知》是政办发〔 2012 〕 24 号文件规定：“省、市州两级人力资源和社会保障部门负责管理本地区社会保障卡发行和应用工作，其所属的信息化综合管理机构具体承担社会保障卡发行和技术管理的有关事务；其所属的就业服务机构、人才服务机构、社会保障经办机构等工作机构，具体承担社会保障卡的应用工作。1、省级人力资源和社会保障部门负责制订全省社会保障卡建设规划、管理办法和标准规范；负责管理全省卡密钥，制作发放全省PSAM卡，实行全省统一初始化；指导各地开展社会保障卡应用，协调社会保障卡省内跨地区用卡业务；2、人民银行A市中心支行负责指导、协调辖内银行机构开展西地省社会保障卡金融应用业务，并对其发行的具有金融功能的社会保障卡进行发卡技术标准符合性审核；3、市州人力资源和社会保障部门按照全省统一规划负责制订本地区社会保障卡建设规划和提出发行注册申请；负责制定本地区社会保障卡基础数据的采集和比对方案，协调本地区社会保障卡的发行管理和应用服务，配合部、省开展跨地区用卡业务。” 《西地省人力资源和社会保障厅关于加快推进我省社会保障卡应用的实施意见》（《政府发文》54号）中对省厅、市州和社保卡合作银行的职责规定如下：“（一）省厅统筹协调，整体推进全省社会保障卡应用工作。1、统一部署。建立和完善相关规章制度，统一全省社会保障卡数据采集接口规范、应用系统接口规范、数据交换规范、社会保障卡应用规范、社会保障卡读写规范、补换卡的标准和流程等；加强省级制卡中心和卡管系统建设，完成卡管系统的全省部署；建立和完善省级人力资源基础信息库；协调全省各合作银行完善银行系统与卡管系统接口；2、加强指导，开展各项培训工作，加强对市州的指导，督办市州开展数据采集、发卡和业务系统改造，督促各地完成业务经办流程调整和读卡设备采购部署，建立定期调度与定期通报制度，确保各市州按照全省统一部署，按时按质按量完成任务；3、完善本级。加强省集中系统建设，改造省本级社会保障卡应用的软件硬环境，进一步调整和优化全省各类业务经办流程，完成省本级读卡设备采购和部署，确保省本级社会保障卡应用。（二）市州立足本地，以“全面落实社会保障卡应用”为目标，逐年稳步开通社会保障卡应用项目。4、密切配合。积极配合省厅完成省级人力资源基础数据库、卡管系统以及跨地区数据交换等全省统一性项目建设，及时发现并上报工作中的遇到的新情况、新问题，提出解决问题的建议和意见；5、普及发卡。配合合作银行及时将制好的社会保障卡发放到位，扩大持卡人群覆盖面；6、推动用卡。根据省厅统一的标准、规范，结合本地实际，加强人力资源社会保障业务专网和网络中心建设；完成本地应用系统社会保障卡接口和经办业务系统改造；完善经办业务流程，完成读写终端的安装部署，确保社会保障卡能够真正用起来。（三）合作银行金融急行，以“全面确保金融应用”为目标，发挥优势，迅速扩大社会保障卡使用活跃度。7、加快发卡速度。在各级人力资源社会保障部门配合下，积极组织移动服务点发卡，确保在收到成品卡1个月内将卡发放至个人手中；8\改善用卡环境，确保各地区发卡前读写终端全部改造到位；完成卡管系统接口改造和联调工作；实现社会保障卡自助查询功能；充分利用银行网点分布广泛的优势，使社会保障卡具有最广泛的应用环境，提高活跃度；9、通力合作共赢。积极配合各级人力资源社会保障部门开展工作，鼓励合署办公，方便群众办事，形成多方共赢局面。” 人社部、人民银行、省人社厅相关文件并未对社保卡加载银行卡功能后的银行短信服务收费做另行规定，且该短信服务费是银行根据相关政策、法规采取的商业收费行为，人社部门没有收取任何费用。？二、约谈相关发卡银行?我们已经正式通知社保卡合作银行邮储银行，并和邮储银行组织召开多次协调会议，经仔细研究诉求人的诉求要点，责成邮储银行一定要严格按照人社部、中国人民银行所下发文件规定的依据进行社保卡的发放与应用。？三、对于您所提的“希望人社部门准许退休人员选择社会保障卡办理银行或准许转行办理社会保险卡。”的问题。F市社保卡合作银行目前有八家，退休人员可以自行选择社保卡发卡银行，如果您有中意的银行，是可以申请变更社保卡发卡银行的。 如有其他疑问，欢迎你拨打服务电话。 F市社会保障卡服务中心电话：8251931。

图 6-1 附件 4 第 1740 条答复意见

6.2 方案算法流程

针对问题三设计了一种基于卷积神经网络(CNN)的答复质量评价方案，该方案的主要部分为以下三点：第一，对答复意见进行基于留言详情的分类，方法参考问题一，并人工标注数据集的答复质量类别；第二，通过 TF-IDF 进行关键词的提取

并保存为词典，作为后面数据集的特征选择指标；第三，拆分特征选择后的数据集放入卷积神经网络内训练，进行答复质量分类。方案算法流程图如下图 6-2 所示：

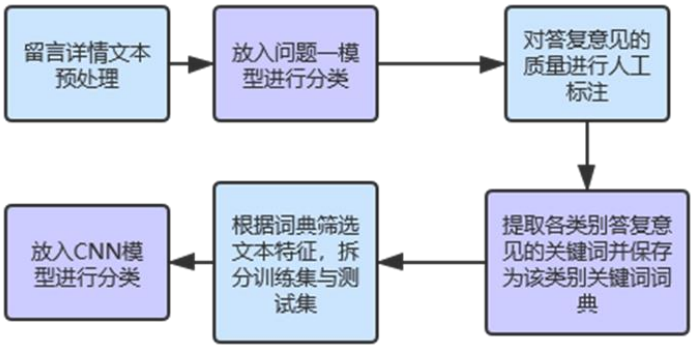


图 6-2 方案算法流程图

对数据预处理及留言详情分类不再过多描述，下面主要是对人工标注答复意见的评价标准、基于 TF-IDF 的关键词提取算法及基于词袋模型的语义特征选择的介绍。

6.2.1 人工标注的评价标准

实验数据使用附件 4 的答复意见这一列，训练集的标注质量决定了分类结果的准确性。本方案的数据集采取如下办法构造：

进行人工标注时，需要将类别分得足够细致才能确保标注的准确性，因此将答复意见的质量分为优质答复和一般答复，而优质答复又分为 A 和 B 两类，一般答复分为 C 和 D 两类，共 4 类。评价标准如下表 6-1 所示：

表 6-1 对答复意见的评价标准

答复质量		评价标准
优质答复	A	有具体的行动并使问题得到改善
	B	有具体行动但问题改善不明显或只针对其中部分问题有行动

一般答复	C	有行动计划
	D	无解决方案或答非所问

由上面表格可以知道，答复质量按照 A～D 依次递减。

本方案以交通运输这一类别为例，按照人工标注后的结果如下图 6-3：

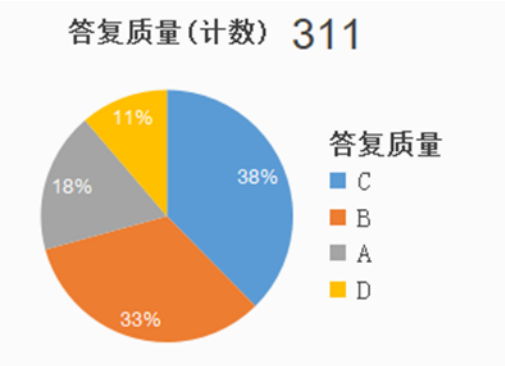


图 6-3 交通运输答复质量计数

6.2.2 基于 TF-IDF 的关键词提取

作为信息检索与数据挖掘的常用加权技术，TF-IDF 常被用来评估一个词语对于文本库中单个文件的重要程度。词语在文本中出现的次数越多，词语重要性越强，但是随着在文本库中其他文本出现频率增加，词语重要性逐渐降低[12]。TF(Term Frequency)是词频的简称，可理解为文本内词语出现的频率，逆文本频率的缩写为 IDF（Inverse Document Frequency），即一个词语普遍关键性的度量。

此模型的核心思想为：若某词语于一个文本内多次出现，即 TF 较高，同时甚少出现于其他文本内，那么判定该词语具备良好类别区分性能，在分类方面具备适用性。按照 IDF 公式得到的 IDF 的值会小，就说明该词语类别区分能力不强。然而客观情况是，若一个词语频繁出现于一个类的文本内，那么表示此词语可很好体现这个类的文本属性，对于此类词语，应赋予其较高权重，同时可将其当作此类文本的特征（关键）词，用来和其他类文本作鉴别。

TF-IDF 三个值 TF、IDF、TF-IDF 的计算公式如下：

$$TF_w = \frac{\text{在某一类词条}w\ \text{出现的次数}}{\text{该类中所有词条的数目}} \tag{1}$$

$$IDF = \log \frac{\text{语料库的文档总数}}{\text{包含词条}w\text{的文档数}+1} \quad \text{（加1避免分母为0）} \tag{2}$$

$$TF\text{-}IDF = TF * IDF \tag{3}$$

在该方案中，对七个类别的答复意见都进行了基于 TF-IDF 算法的关键词提取，并计算了它们的 TF-IDF 权重，保存于 task3 的 keywords 文件夹中。通过程序绘制了交通运输这一类别的关键词如下图 6-4：

图 6-4 交通运输关键词词云

6.2.3 基于词袋模型的语义特征选择

词袋模型是用于描述文本的一个简单数学模型，也是常用的一种文本特征提取方式。在信息检索中，词袋模型假定对于一个文本，忽略其次序和语法，仅仅当作是该文本中若干个词语的集合。该文本中，每个词语都是互不相关的，每个词语的出现都不依赖于其他词语。也就是说，文本中任意一个词不管出现在任意哪个位置，都不会受到其他因素的影响[13]。

要构建数据集，就要先将文本转换为数值格式，以便进行机器学习，分析数据并且提取有用信息。基于上面构建的关键词库，词袋模型从文本中提取出词库中出现的可以代表其文本特征、具有语义信息的关键词。并且用这些关键词构造特征向量。这就使得每一份文本可以描述成一个词袋，被描述成各种词语权重的组合体。

而且只需要记录词语的数量，语法和单词的顺序都可以忽略。如图 6-5 这个答复，只保留了对语义分析有用的关键词：

'网友 您好 您 的 留言 已 收悉 现将 有关 情况 回复 如下 根据 对区 石坝 路 马王堆 路口 交通标志 标线 实地考察 后市 交警支队 初步 研究 认为 根据 中华人民共和国 道路交通 安全法 实施 条例 第三十八条 第一款 第三项 红灯 亮时 禁止 车辆通行 之 规定 左 转弯 信号灯 为 红灯 时 禁止 车辆 越过 停止 线 行驶 如 您 对 处罚 结果 有 异议 可 向 当地 交警部门 申请 复议 感谢您 对 我们 工作 的 支持 理解 与 监督 年月日'

图 6-5 特征选择前

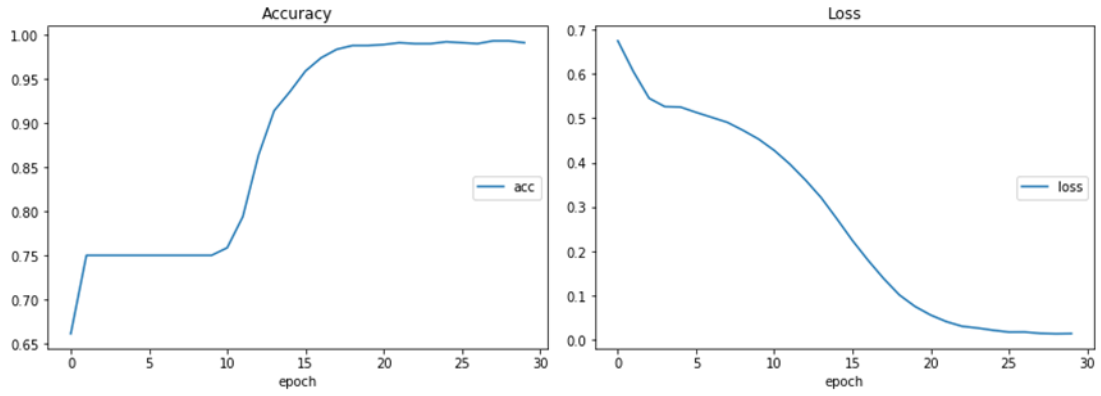
{'交警支队': 0.327, '中华人民共和国': 0.327, '行驶': 0.32, '道路交通': 0.314, '交警部门': 0.314, '处罚': 0.303, '研究': 0.29, '路口': 0.265, '申请': 0.251, '实施': 0.222, '车辆': 0.194, '监督': 0.178, '理解': 0.177, '现将': 0.167}

图 6-6 特征选择后

6.3 实验结果及分析

6.3.1 实验结果

将处理过后的数据集放入 CNN 卷积神经网络。当 epochs 为 30，batch_size 为 64 时，训练集每次迭代的准确率如下图 6-7 左所示，通过设置最大迭代次数进行模型训练终止条件，每次训练结束后，都进行准确率评估，横坐标是迭代次数，纵坐标是预测结果的准确率；迭代的损失值如下图 6-7 右所示，横坐标代表迭代次数，纵坐标代表每次迭代的损失值。训练时通过设置准确率阈值和损失阈值或设定迭代次数结束训练。迭代次数过多会造成过拟合现象，降低模型的预测效果。




```
Accuracy
        acc                 (min:    0.737, max:    0.997, cur:    0.994)
Loss
        loss                (min:    0.009, max:    0.663, cur:    0.011)
```

图 6-7 模型训练的准确率与损失率

可见，经过 10 次迭代后，训练准确率稳步上升，20 次之后趋于稳定。而损失率一开始是 66.3%，最低趋于 0。

计算 A、B、C、D 的精确率(Precision)、召回率(Recall)及 F1-score 这些可以评价分类模型的得分,如下表 6-2 所示：

表 6-2 模型的评估得分

答复质量类别	precison	recall	F1-score
A	0.2917	0.3182	0.3043
B	0.5833	0.7778	0.6667
C	0.2857	0.1333	0.1818
D	0.5429	0.5938	0.5672
Avg	0.4259	0.4558	0.4300

可见，B 类的 F1-score 最高，C 类的最低，四个类别的均值为 0.4300，均不超过 0.5。

6.3.2 分析与改进

4 个答复质量类别中的 F1-score 均值不超过 0.5，说明分类结果并不理想，经分析有以下几个原因：

- 本方案通过文本自身所包含的信息进行答复意见质量评价，但是只考虑了语义特征，未考虑文本的结构化特征（如标点符号占比、平均词长等）及语言学特征（如词性、句法等）；
- 对留言所属一级标签的分类正确率不高；
- 数据量少，且留言中问题发生的地点较为集中，提取关键词时未过滤掉地点名词，导致分类结果不理想；
- 人工标注的答复质量类别容易出现疏漏及误判。

根据方案的设计，可以提出以下**改进**方法：

- 可以将文本的结构化特征（如标点符号占比、平均词长等）及语言学特征（如词性、句法等）这些特征加入以提高评价分类的准确率。

- 从实验结果看，留言所属一级标签的分类错误对关键词获取和答案质量评价有严重干扰，未来可以通过更高效的分类算法标注留言所属话题类型。
- 增加答复意见的数据量，并且人工标注答复意见类型要进行复查，避免标注错误和遗漏.
- 可以通过改进网络结构提高分类准确率，例如通过运用典型梯度下降（gradient descent）优化方法和注意力机制多特征融合等。

7 总结与展望

各类社情民意相关的文本数据量不断攀升，给以往主要依靠人工来进行留言划分和热点整理的相关部门带来极大挑战。依靠人工处理，存在工作量大、效率低，且差错率高等问题。

针对此问题，我们首先根据留言内容（留言主题和留言详情）的特征对留言使用 TextCNN 卷积神经网络模型进行分类，减少人工工作量；在热点整理方面，我们使用了基于 DBSCAN 模型对热点问题进行聚类，然后结合 TextRank 算法和统计对热点话题进行了排序，方便相关部门及时处理热点问题；最后，为了对答复意见质量进行划分，建立了一个基于卷积神经网络 CNN 的答复质量评价方案，以便政府部门改进工作。

对三个问题的解答中，我们借鉴了比较主流的文本机器学习方法，如卷积神经网络 CNN 及 DBSCAN 聚类，得出了较为理想的结果。本文还需要对比更多其他的方法或优化网络结构来提高分类或评价的正确率，这是本文可以更完善的方向。

8 参考文献

[1] 王翔. 基于数据挖掘的热点新闻发现及系统方法研究[D].湖北工业大学,2017.

[2] 王馨. 网络新闻热点发现研究[D].河北大学,2015

[3] 陈龙. 新闻热点话题发现及演化分析研究与应用[D].南京理工大学,2017.

[4] 侯小培,高迎.卷积神经网络 CNN 算法在文本分类上的应用研究[J].科技与创新,2019,(4):158-159. DOI:10.15913/j.cnki.kjycx.2019.04.158.

[5] 万磊,张立霞,时宏伟.基于 CNN 的多标签文本分类与研究[J].现代计算机,2020(08):56-59+95.

[6] 北京工业大学.一种基于 TextCNN 改进的文本分类方法:CN201910174176.1[P].2019-06-21.

[7] https://blog.csdn.net/huacha__/article/details/81094891

[8] 叶健成, 利用文本挖掘技术进行新闻热点关注问题分析

[9] https://www.cnblogs.com/clover-siyecao/p/5726480.html

[10] 肖天久, 刘颖. 基于聚类和分类的金庸与古龙小说风格分析[J]. 中文信息学报, 2015,29(5):167-177.

[11] https://baike.baidu.com/item/%E8%AF%AD%E4%B9%89%E4%BF%A1%E6%81%AF/5180120?fr=aladdin

[12] 张波,黄晓芳.基于 TF-IDF 的卷积神经网络新闻文本分类优化[J].西南科技大学学报,2020,35(01):64-69.

[13] 黄春梅,王松磊.基于词袋模型和 TF-IDF 的短文本分类研究[J].软件工程,2020,23(03):1-3.

9 附录

```
#卷积神经网络 CNN 函数

def CNN(input_dim,
        input_length,
        vec_size,
        output_shape,
        output_type='multiple'):
    """
    Creat CNN net,use Embedding+CNN1D+GlobalMaxPool1D+Dense.
    You can change filters and dropout rate in code..

    :param input_dim: Size of the vocabulary
    :param input_length:Length of input sequences
    :param vec_size:Dimension of the dense embedding
    :param output_shape:Target shape,target should be one-hot term
    :param output_type:last layer type,multiple(activation="sigmoid") or single(activation="softmax")
    :return:keras model
    '''
    data_input = Input(shape=[input_length])
    word_vec = Embedding(input_dim=input_dim + 1,
                         input_length=input_length,
                         output_dim=vec_size)(data_input)
    x = Conv1D(filters=128,
               kernel_size=[3],
               strides=1,
               padding='same',
               activation='relu')(word_vec)
    x = GlobalMaxPool1D()(x)
    x = Dense(500, activation='relu')(x)
    x = Dropout(0.1)(x)
    if output_type == 'multiple':
        x = Dense(output_shape, activation='softmax')(x)
        model = Model(inputs=data_input, outputs=x)
        model.compile(loss='binary_crossentropy',
                      optimizer='adam',
                      metrics=['acc'])
    elif output_type == 'single':
        x = Dense(output_shape, activation='sigmoid')(x)
        model = Model(inputs=data_input, outputs=x)
        model.compile(loss='categorical_crossentropy',
                      optimizer='adam',
```

```
                  metrics=['acc'])
    else:
        raise ValueError('output_type should be multiple or single')
    return model


if __name__ == '__main__':
    model = CNN(input_dim=10, input_length=10, vec_size=10, output_shape=10, output_type='multiple'
)
    model.summary()
```

```
#留言聚类函数

def get_cluster(matrix, cluster='DBSCAN', cluster_args=None):
    """
    对数据进行聚类，获取训练好的聚类器
    :param matrix: 稀疏矩阵
    :param cluster: string，聚类器
    :param cluster_args: dict，聚类器参数
    :return: 训练好的聚类器
    """
    cluster_args = {'eps': 0.5, 'min_samples': 5, 'metric': 'cosine'} if cluster_args is None else cluster_args
    cluster_args_list = ['%s=%s' % (i[0],
                                    '"%s"' % i[1] if isinstance(i[1], str) else i[1]
                                    ) for i in cluster_args.items()]
    cluster_args_str = ','.join(cluster_args_list)
    cluster1 = eval("%s(%s)" % (cluster, cluster_args_str))
    cluster1 = cluster1.fit(matrix)
    return cluster1

def content_cluster(df, df_save=False):
    """按留言内容聚类"""
    df_content = df.copy()
    df_content = content_preprocess(df_content)
    word_library_list = get_word_library(df_content['content_cut'])
    single_frequency_words_list = get_single_frequency_words(df_content['content_cut'])
    max_features = len(word_library_list) - len(single_frequency_words_list) // 2
    content_matrix = feature_extraction(df_content['content_'], vectorizer='CountVectorizer',
                                        vec_args={'max_df': 0.95, 'min_df': 10, 'max_features': max_features})
    content_DBSCAN = get_cluster(content_matrix, cluster='DBSCAN',
                                 cluster_args={'eps': 0.35, 'min_samples': 4, 'metric': 'cosine'})
    content_labels = get_labels(content_DBSCAN)
    df_content['content_label'] = content_labels
    df_non_outliers = get_non_outliers_data(df_content, label_column='content_label')
```

```
    content_label_num =  get_num_of_value_no_repeat(df_non_outliers['content_label'].tolist())
    print('按留言内容聚类，一共有%d 个簇(不包括离群点)' % content_label_num)
    content_rank =  label2rank(content_labels)
    df_content['content_rank'] = content_rank
    for i in range(1, content_label_num + 1):
        df_ = df_content[df_content['content_rank'] == i]
        content_top_list =  get_most_common_words(df_['content_cut'], top_n=15, min_frequency=1)
        print(content_top_list)
    if df_save:
        df_content.drop(['content_', 'content_label'], axis=1, inplace=True)
        save_news(df_content, os.path.join(results_path, 'df_content_rank.csv'))
    return df_content
```