“智慧政务”中的文本挖掘应用

摘要

近年来，随着电子政务的发展，各类社情民意相关文本不断涌现。政务服务完善的过程伴随着工作量的攀升，条条留言中隐含着民意民情与亟待解决的问题。自然语言处理（NLP）作为人工智能的重要领域发展迅速，可作为有效工具助力于政务处理，提高施政效率，及时了解群众，关心民情。

针对具体问题，从以下三个方面进行阐述：

对于问题一；首先对各种文本表示方法与机器学习算法进行了组合实验，并对深度学习模型在群众留言分类进行实验。结果发现机器学习中基于 FastText-XGBoost 和 TF-IDF-SVM 算法在文本分类中表现出色，深度学习中 TextCNN，DPRCNN，Fasttext 方法具有良好的表现。随后，进一步对分类模型提出优化，对于机器学习应用 CPSO 算法和 IAPSO 算法对 XGBoost 和 SVM 模型进行优化，其 F1 达到 0.845329 和 0.871915，对于深度学习模型提出了一种带类别权重的卷积神经网络模型进行改良。优化后的模型提高了分类准确率达到 0.910966341，F1 值提升到 0.906972567。

对于问题二，首先采用基于 Singlepass 算法与基于有监督的 LDA 模型的划分法做不同的话题划分；其次，经过对比，选用效果更优的基于 LDA 的话题划分结果；再而，提出热度计算公式。最终选取热度值最高的前五个话题作为第二题结果。

对于问题三，首先构建答复文本优良特征，从内容，格式，合理程度，时效四个角度分析；其次，采用线性回归分析方法对文本风格进行分析；接着采用四种聚类算法对答复文本优良特征进行分析。经过对比，最终选用 k-means 聚类算法的结果对答复文本优良程度进行评价。

关键词：电子政务；留言分类；CPSO；IAPSO；XGBoost；SVM；类别权重的卷积神经网络模型；热点挖掘；Singlepass；LDA；答复评价模型；回归分析；k-means

Abstract

In recent years, with the development of e-government, various texts about social conditions and public opinion have emerged. The process of perfecting government services is accompanied by a rise in workload, and all messages imply public opinion and problems that need to be resolved. Natural language processing (NLP), as an important field of artificial intelligence, is developing rapidly. It can be used as an effective tool to assist in government affairs processing, improve governance efficiency, understand the masses in a timely manner, and care about the public.

For specific issues, the following three aspects are explained:

For question one; firstly, a combination experiment of various text representation methods and machine learning algorithms is carried out, and an experiment of deep learning models in the mass message classification is conducted. It is found that the algorithms based on FastText-XGBoost and TF-IDF-SVM in machine learning perform well in text classification, and the TextCNN, DPRCNN, and Fasttext methods in deep learning have good performance. Subsequently, the classification model was further optimized. For machine learning, the CPSO algorithm and IAPSO algorithm were used to optimize the XGBoost and SVM models. Its F1 reached 0.845329 and 0.871915. For the deep learning model, a convolutional neural network model with category weights was proposed. Make improvements. The optimized model improved the classification accuracy to 0.910966341, and the F1 value was increased to 0.906972567.

To solve the second problem, firstly, the division method based on the Singlepass algorithm and the supervised LDA model is used to divide different topics; secondly, after comparison, the LDA-based topic division result with better effect is selected; and then, the heat calculation formula is proposed. Finally, the first five topics with

the highest heat value are selected as the result of the second question.

For question three, first construct the excellent features of the reply text, and analyze it from the four perspectives of content, format, reasonableness, and timeliness; secondly, use linear regression analysis to analyze the text style; then use four clustering algorithms to analyze the excellent features of the reply text For analysis. After comparison, the results of k-means clustering algorithm were finally selected to evaluate the excellentness of the reply text.

KEYWORDS：E-government; message classification; CPSO; IAPSO; XGBoost; SVM; category weighted convolutional neural network model; hot spot mining; Singlepass; LDA; reply evaluation model; regression analysis; k-means

目录

图录

表录

第一章 引言

1.1 研究背景

新一代信息技术的加速驱动下，我国政府不断谋求职能转变和效率提升，驶入了电子政务建设的快车道。20 世纪 80 年代，我国开始探索办公自动化，90 年代“三金”工程拉开了在线政务服务平台建设的序幕，尤其是国家深入推进“互联网+政务服务”改革，各级各地纷纷建设地方和部门在线政务服务平台，显著提高了工作效率，深受群众欢迎。2018 年 7 月，国务院印发《关于加快推进全国一体化在线政务服务平台建设的指导意见》，2019 年 11 月，全国一体化在线政务服务平台上线试运行。

电子政务领域的应用也从早期诸如门户网站、邮件系统等简单应用，不断得到创新和拓展，网上便民服务平台、掌上政务、智慧城市等各种业务支撑系统如雨后春笋般相继涌现。电子政务作为政府在新时代信息技术运用过程中政府服务方式的一种转变，在政府的日常工作中发挥着重要作用。

1.2 研究意义

近年来，我国经济快速发展，2018 年我国 GDP 超过 90 万亿元，市场主体已超过 1 亿户；城市化进程持续加快，城市人口从 1978 年占全国的 18%增长到 54%，近 90 个城市城区人口超过百万。城市人口聚集和市场主体规模扩大为各地创新政府治理提供了新动力的同时，也给社会治理和公共服务带来新挑战。

自 2019 年 5 月国家政务服务平台全面上线试运行以来，平台累计访问浏览量越 2.22 亿次、实名注册用户 1049.6 万，为地方部门提供实名身份核验服务 3358 万次、电子证照调用共享服务 286 万次。此外，我国 31 个省（区、市）及新疆生产建设兵团和 40 多个国务院部门已全部开头网上政务服务平台，共计 15143 个，包括政府门户网站和部门网站。其中，国务院部门及其内设、垂直管理机构共有政府网站 1001 个；省级一下行政党委共有政务网站 14142 个，分布在我国 31 个省（区、市）及新疆生产建设兵团。

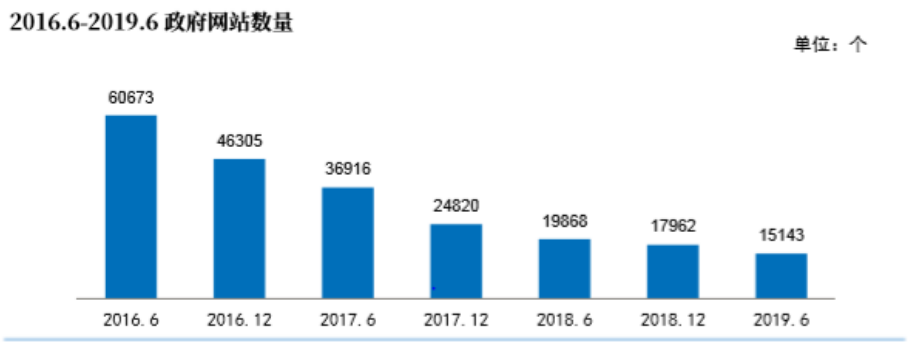


图 1 政府网站数量

Figure 1 Number of Government Website

截至 2019 年 6 月，我国在线政务服务用户规模达 5.09 亿，占网民整体的 59.6%；已有 2297 个地级行政区政府已开通了“两微一端”等新媒体传播渠道，总体覆盖率达 88.9%。

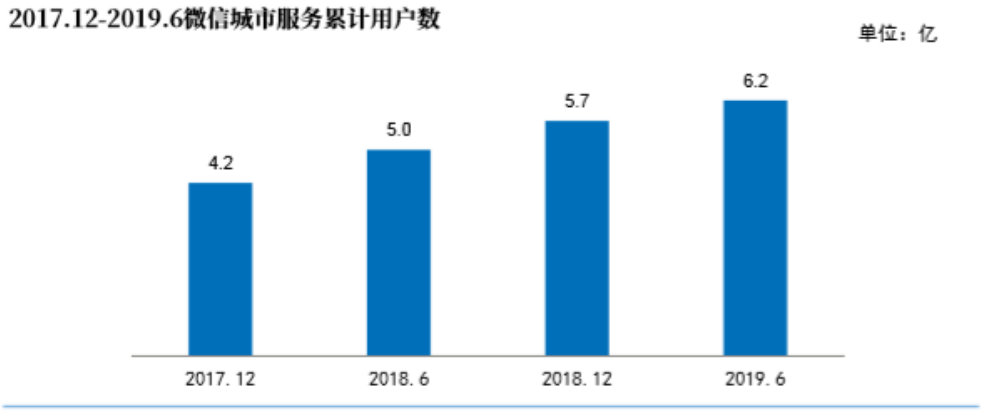


图 2 微信城市服务用户数

Figure 2 WeChat city service users

在线政务平台的发展，服务需求的激增，使得仅靠人工对群众留言进行划分与整理的工作方法已不能满足智能化需求。 运用自然语言处理、文本挖掘等相关技术对留言或提问进行智能分析，如确定留言主题，留言分类，热点整理，以及通过分析留言，找出人民关注点的变化等，可以协助工作人员对留言更深入的了解和掌握，减少人力负担，进一步提高办公效率。

为此，针对政务留言平台，设计如下方案，自动进行留言分类，热点提取，以及对回复质量做评价，是具有重要意义的。

第二章 数据预处理

观察题目中所给予的原始数据，发现原始数据中存在大量冗余、不一致的信息以及非结构化、异常的数据。此类数据将会严重影响数据挖掘和建立模型的执行效率，甚至会导致所建模型和根据数据挖掘产生的结果出现较大偏差，故数据预处理在建模的步骤中显得尤为重要。为了提高数据集的质量并且让数据更好地适应所建立的数学模型，现对题中给定的原始数据进行操作。

对文本数据进行预处理[1]是文本表示建模的第一步，其处理效果的好坏直接影响后续所有的操作效果，文本预处理步骤如图 3 所示。

图 3 数据预处理流程

Figure 3 Data Preprocessing

2.1 数据清洗

数据清洗就是清理脏数据以及净化数据的环境。一般来说，脏数据就是原始数据中无关数据、重复数据和有噪声的数据，如：‘\u3000‘,’ \t ‘,’\n‘,’ \xa0‘。本节给出清洗过程中如何处理这些问题的方法。本文字符和符号的过滤采用的是 Python 程序中的 re 库实现正则表达式的过滤，数据清洗模型如下：

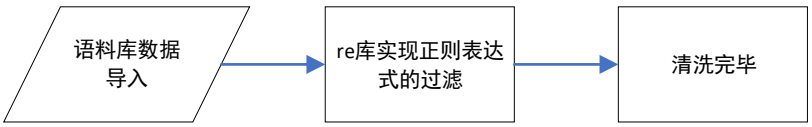


图 4 数据清洗模型

Figure 4 Data Cleaning Model

2.2 分词处理

对于英文文档而言，词与词之间的分隔是通过特定的间隔标记符号实现的，例如空格和标点符号等。所以遍历文档，就能够实现英文文档的分词，并获得单词列表。而对于中文文本，分词处理是一个很重要的研究方向，目前已有很多相对成熟的中文分词工具：

(1)由中国科学院计算技术研究所研发的 ICTCLASS 中文分词系统[2]。该系统基于层叠隐马尔科夫模型，具有百万级别的词典库，具有中文分词、词性标注、命名实体识别、新词识

别等功能。

(2)IKAnazer[3]是一个轻量级的 Java 中文分词工具。它采用了特有的“正向迭代最细粒度切分算法”和多子处理器分析模式，支持用户词典扩展定义，分词准确率较高。

(3)jieba 中文分词系统是当前被广泛应用的 Python 分词工具。jieba 基于前缀词典实现高效的词图扫描，找出语句中汉字所有可能生成词情况所构成的有向无环图，再采用动态规划查找最大概率路径，找出基于词频的最大切分组合，此方法具有较好的分词效果。

通过对上述三种分词工具的分析，考虑到三种工具都相对成熟，而 jieba 中文分词器分词效果略胜一筹且操作简单，本文采用了 jieba 分词工具对文本数据进行分词处理。

2.3 数据过滤

在自然语言处理中，数据过滤主要指对停用词和低频词的过滤，停用词和低频词会给文本处理带来极大干扰，因此在文本分词之后需要进行数据过滤，这样一方面降低了噪音干扰，另一方面降低了文本向量维度，从而降低建模难度。

2.3.1 低频词

低频词，顾名思义就是在数据中出现次数较少的词语。此类数据实际上是具有一定的信息量，但是把低频词放入模型当中运行时，它们常常保持他们的随机初始状态，给模型增加了噪声。对于低频词的处理，简单的方法就是移除他们。低频词表生成模型如下：

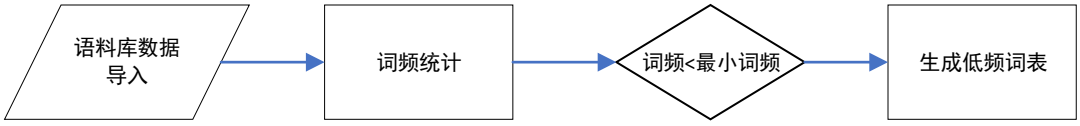


图 5 低频词表生成模型

Figure 5 Model of Low Frequency Vocabulary Generation

2.3.2 停用词

停用词(stop words)又称功能词(function words),通常指在各类文档中都频繁出现,因而被认为很少带有有助于分类的任何信息的代词、介词、连词等高频词。去停用词(stop words removal)的目的是将停用词从文本表征词典中剔除掉。

在自然语言处理中，停用词通常指没有实际含义的虚词，主要包括了语气词、副词、拟声词、介词、连词、叹词等等。这些类型的词汇通常没有明确的词义，只有在完整的语境之

中才具备一定的语法意义[4]，如前面这段话中的“并”、“也”、“的”、“才”等等。一般认为，这些类型的词所附着的文本信息较少，对于文本分类的贡献不大，可选择将其从中文分词的结果中过滤掉。 罗杰等人[5]的研究认为，停用词不仅包括许多类型的虚词，还包括实词中的数词、量词、代词、方位词等，以及一些无实际意义的动词和过于常用的名词。在分词的过程中，剔除这些类型的词汇对文本主干信息的提取没有太大影响，而且有利于消除歧义。同时，周钦强等人[6]的研究显示，在文本分词的过程中，存在大量的独立单字以及含有数字符或英文符的中文词汇。它们所携带的文本信息量通常较小，并且容易给其它的词汇带来一定的抑制作用。因此在文本分类的过程中，也可以过滤掉这些类型的字或词。 总之，停用词通常是那些不具有重要的文本信息价值的字或词汇，然而对于某个给定的目的而言，任何类型的词汇都可以被选作停用词。

实际应用时，研究者通常会根据研究内容来设定停用词，以此达到期望的分词效果。目前主流的通用中文停用词表有百度停用词表、哈尔滨工业大学停用词表以及四川大学机器智能实验室停用词表，其统计情况如表 1 所示。

表 1 实验停用词表内容统计

Table 1 Statistics of Experimental Stop Word List

停用词表	符号	英文	单字词	两字词	三字词	四字词	其他	共计
百度	7	547	173	620	29	19	0	1395
四川大学	0	0	26	663	80	84	6	859
哈尔滨工业大学	236	0	167	290	23	19	0	750

表 2 显示了三个停用词表的重合情况,可以看出,这三个停用词表的单字词、三字词及四字词的重合率很高,基本达到 80%以上,因此三个停用词表的区别主要体现在两字词上。

表 2 停用词表重合词条统计

Table 2 Statistics of Overlapping Entries in Stop Lists

对比词表	单字词	两字词	三字词	四字词	共计
百度－四川大学	22	311	23	19	374
百度－哈尔滨工业大学	167	288	22	18	493
四川大学－哈尔滨工业大学	22	276	22	18	338
百度－四川大学－哈尔滨工业大学	22	275	22	18	337

针对本题,合并了三个停用词表作为一个新的停用词表，对文本信息进行数据过滤。

第三章 文本表示模型

随着互联网的兴盛产生了大量的信息，文档成为信息存储的最通用的形式。互联网是产生电子文档的最多的来源，例如网页、邮件、微博、微信等等，每一个信息源都可能包含上亿的文本。一般来说，文本形式的数据是自然的、非结构化的，对文本进行文本表示的模型一般都是高维的、语义的、稀疏的，那么如何设计一个好的文本表示模型成为文本分类问题中的一个挑战。文本分类任务的定义是把文本分类到预先定义好的一个或多个类当中。文本分类广泛应用于意见融合、情感分析（检测评论或者推特的语义极性）[7]、话题分类、热点分析、文本评价。在自然语言处理中，特征词是文本表示最佳单元[8]。尽管文档可以表达更多的信息，但是文档没有像传统的数据库一样的组织结构，文本形式的数据可以表达非常广泛的信息，但是这种形式的信息编码很难从计算机中进行自动解释，那么我们需要把这种非结构化的文本信息转换成计算机可以读懂的形式，为了实现这个目的，现有的文献中提出了很多预处理的算法[9]。在把非结构化的文本转换成结构化的文本之后，我们需要提供一个有效的文本表示模型，为建立强有力的自然语言处理系统提供支持。

3.1 布尔模型（Boolean model）

布尔模型是基于特征项的严格匹配模型。首先，需要产生一个二值的变量，这些变量对应着文本的特征项。这些特征项构成向量来表示文本。如果文本包含有这个特征项上的特征词，那么该特征项上的值为 1 或者“true；反之为 0 ，或者“false”。对于文本的查询是由文本的特征项和逻辑操作的“AND”、“OR”、 “NOT”组成。查询过程中的匹配原则需要遵循布尔计算原则。布尔模型是在 60 和 70 年代间诞生的[10]。当时有很多的商业信息检索系统都采用了这种模型，例如：DIALOG、STAIRS 和 MEDLARS 等。使用布尔模型的主要优势就是检索高速以及表示结构化文本的便易性。布尔模型的缺点是表达不精确，不能够表示出文本中那些重要的特征项，不能为文本挖掘提供有灵活的、有质量的分析，这样会导致在检索中匹配过于宽泛，用户真正需要检索的文本会被遗漏。

3.2 空间向量模型（Vector Space Model）

词袋模型（Bag of Word，BoW）是文本表示中的一种基本模型。在 BoW 模型中通过特征词在文本中出现的频率来构建文本特征向量，特征向量中的每一维对应着相应的特征词的频

率，这种文本表示模型又称为向量空间模型（Vector Space Model，VSM）。

VSM 模型是 Salton 在 1968 年提出的，它是信息检索领域一个经典的语言计算模型[11]。在 VSM 模型中，用文本中的每一个词去表示文本的特征。每一个文本被表示成向量空间中的一个点，这个点是由一组归一化的正交的特征向量组成的，即把文本视为在 n 维空间中的一个向量（模型假设：文本中特征词的出现是独立的，相互之间没有关联和影响）。文本被表达成如下形式：

$$d = (w_1, w_2, ..., w_n)$$

其中，w_i 是文本 d 的第 i 个特征词在文本中的权重，这个权重是用来描述该特征词在文本中的重要性。权重值越大，那么特征词在文本中就越重要。通过特征词在文本中出现的频率来刻画特征词的重要程度。

特征词频率又分成相关特征词频率和绝对特征词频率。绝对特征词频率是特征词在文本中出现的次数；相对特征词频率是标准特征词频率。计算相对特征词频率通常使用 TF-IDF（Tem Frequency－Inverse Document Frequency）。TF-IDF 的计算公式 Salton 和 McGill 在 1983 年提出[12]。这种特征词频率的计算方法是受到向量空间信息检索模式的启发。 T F 表示特征词出现的频率，即特征词在给定文本中出现的次数。IDF 是逆频率，它表示的是在一个文本集中特征词的统计频率。TF-IDF 的计算公式如下：

$$W(T,D) = \frac{tf(t,D) \times log(N/n_i + 0.01)}{\sqrt{\sum_{i=1}^{n}[tf(t,D) \times log(N/n_i + 0.01)]^2}}$$

其中 $tf(t,D)$ 表示特征词 t 在文本 D 中出现的频率。N 为所有训练文本的总数。n 是文本向量的维度。n_I 为特征词 t 出现在所有训练文本中的文档数量。分子是归一化因子，目的是为了保证文本向量的各个向量值的累加和为 1。

在 TF-IDF 文本表示模型中有一个假设，即可以最好的区分文档的特征词一定是在某一类中出现次数最多的同时在其他类别中出现次数一定是最少的。VSM 模型的一个优势就是文本的内容被形式化成多维空间中的一个点，这个点就是一个向量。文本被定义成在该空间中用实值形式表达的一个向量，这样为自然语言的处理提供了可计算性和可操作性。给每一个特征词赋予一个权重，通过调整文本包含的特征词的权重来表示文本一定程度上克服了传统的布尔模型的缺点[13]。VSM 模型的缺点是：1）过度强调了特征词的特性而忽视了特征词的共性；2）仅仅从词义表面来描述文本忽视了特征词之间的相似含义。尽管 VSM 模型是表达文本的一个比较好的方法，但是它是基于一个先决条件的，即文本中的特征词之间是正交的。

这个假设忽视了特征词的上下文，这和自然语言中的实际的文本内容是相矛盾的。因此 VSM 模型无法完全表达出文本的语义。VSM 模型中关于文本独立的假设在自然语言中是不实际的。一个特征词作为语言的一个部分必然会和其他的特征词产生各种相关性，这就导致了 VSM 模型的一个巨大的缺陷。VSM 模型不考虑词语之间的语义关联，也不考虑潜在的概念结构上的关联（例如上下文词语之间的共现和同义）。为了克服这些问题，一些学者提供了很多新的概念和方法。Hotho 等人在 VSM 基础上提出了一种本体文本表示以期能够体现相邻特征词的语义关联[14]。基于本体的模型需要文本向量中的特征词可以充分表达出领域信息，但是这些领域信息的构建需要能够自动构建本体知识为基础，对于缺乏结构化知识的文本而言这样的本体构建是困难的。Cavanar 在文本表示中使用了以字符或者文本为单位的字符串，我们称之为 N-Gram 模型[15]。在 N-Gram 模型中，决定究竟用多少个 gram 来进行有效的文本表示是非常困难的。Wei 等人在 VSM 基础上提出了 LSI（Latent Senmantic Indexing）模型，该模型可以使得文本向量中的每一维对应着有语义相关属性的特征[13]。其他还包括语义向量模型、概念词链等等。

3.3 Word2vec 词嵌入模型

Mikolov 等人在 2013 年提出 Word2vec 模型［２０］，Word2vec 模型［２１，２２］也称为 Word Embedding，中文称之为“词向量”或者“词嵌入”。Word2vec 模型不同于传统的向量空间模型，它最初是提出的一种分布式假设，即在类似情境中出现的单词往往具有相似的含义，基于此假设，使用简单的神经网络将单词嵌入连续的向量空间将词语利用其上下文信息转换成独一无二的多维实数形式表达的向量。在向量空间中，意思表达越相似的词语在向量空间中的距离越近，因此利用该模型计算文本相似度时通常通过计算向量空间上的距离来表示文本语义上的相似程度。Word2vec 模型需要通过训练大量的文本语料，使语料中的每个词都有一个单独的向量表示，包括 CBOW 模型和 Skip-Gram 模型这两种模型。虽作用不同，但都是以 Huffman 树为基础，因此在实际应用中这两种模型的代码也有很多部分是通用的。CBOW 模型（如图 6）是根据上下文的词语 $(w(t-2), w(t+1))$ 来预测当前词语 $w(t)$ 的概率，比如根据：中国的首都是（？）推测目标词语北京；而 Skip-Gram 模型（如图 7）则正好相反，它是从目标字词推测出原始语句。其中 CBOW 模型比较适合用于小型数据，而 Skip-Gram 模型在大型语料中表现的效果更好。

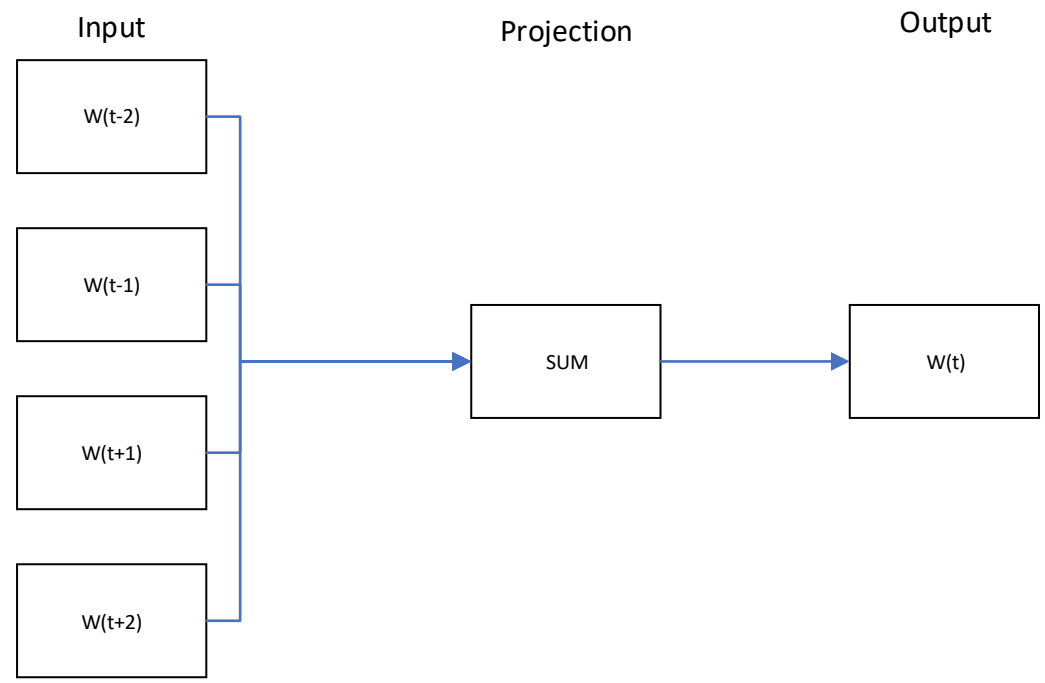


图 6 CBOW 模型

Figure 6 CBOW Model

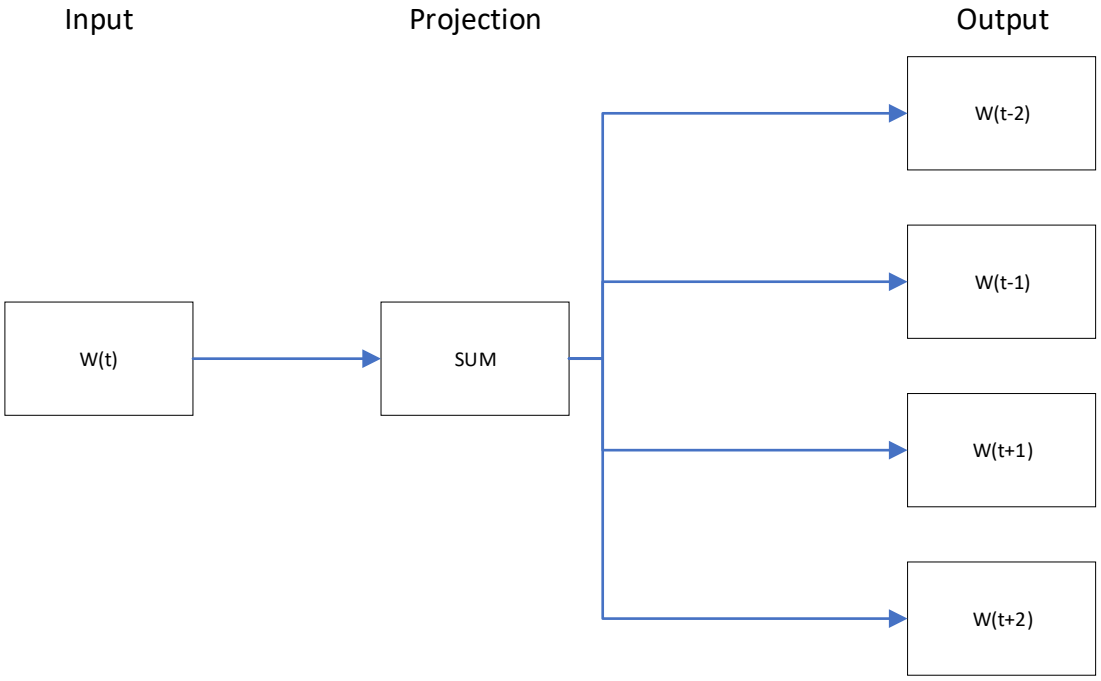


图 7 SKIP-GRAM 模型

Figure 7 SKIP-GRAM Model

Word2vec 模型应用于文本表示主要是对句子分词后得到的词向量进行求和或者取所有词向量总和的平均值得到的向量作为该句子的文本表示。然而这种方法并未考虑到词语在上下文中的实际表达含义，比如“苹果”一词，其作为水果含义的词向量和作为苹果电子设备含义的词向量表示结果完全相同。

3.4 Fast Text 词嵌入模型

Fast Text 是 Word2Vec 的衍生模型，在 Word2vec 模型中把每个单词当成一个单位，从而为每个单词生成一个向量，但 Word2Vec 忽略了单词的内部形态特征，例如“咨询”与“咨询信息”，理论上来理解两个词具有共同的含义，但 Word2Vec 会将它编码成不同的 id，因此会因为这两个 id 不同丢失它的内部结构信息，为了克服这个问题，Fast Text 使用了字符级别的 n-gram 来表示一个单词，并且为了区分前后缀在单词的开头和结尾分别加上“<”,”>”。具体示例例如对于单词“apple”，假设 n 的取值为 2，则它的的 2-gram 特征有“<a”，“ap”，“pp”，“pl”，“le”，“e>”。这种形式的表示方式，对于低频词的训练比较友好，因为可能会跟其他单词拥有共同的词汇特征，且对于不在训练语料之内的词汇可以根据这些字符特征，构建各自的词向量。Fast Text 结构如图 8 所示：

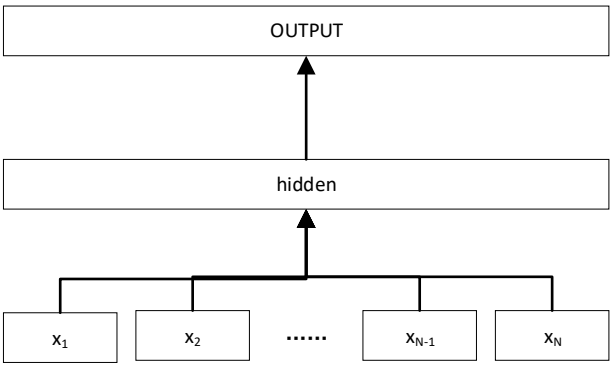


图 8 Fast Text 模型结构图

Figure 8 Fast Text Model Structure

根据上面结构可以发现，同 CBOW 结构相似，Fast Text 的网络结构也是只有三层，和 CBOW 不一样的是，Fast Text 的输入层为初始化的多个单词和它的 n-gram 特征的向量表示，n-gram 特征是单词的额外特征，而 CBOW 的输入为经过 one-hot 编码的单词稀疏向量，两者的隐藏层结构相同，都为词向量的加权平均值，输出层都是某个词在当前文本出现的概率。

3.5 Doc2Vec 词嵌入模型

Word2Vec 和 Fast Text 都是当前主流的词向量表示方式，Word2Vec 是在分词后的单个词汇的基础上进行向量表示，其中每个词汇都有各自的唯一的向量表示，即在不同语境下，同一个词汇的向量表示是唯一的。而 Fast Text 是基于字符级别的词汇表示，能够表示语

料库中未训练的词语，训练速度相比于 Word2Vec 要快很多。Doc2Vec 则是基于段落文章篇幅的词嵌入模型，将文本全局信息考虑进模型，从而避免语境不同时出现的词向量的唯一性。

Doc2Vec 模型是 Le 和 Mikolov 于 2014 年提出的[16]，是一个专门训练词向量的模型，该模型在原 Word2Vec 模型的基础上，在输出层引入段落向量，从而来表示词语的不同含义，对于多义词的表示有了很好的处理。该模型通 Word2Vec 模型类似，也包括两种结构，一种是 Distributed Memory（DM）模型，该模型是 Word2Vec 模型中 CBOW 模型的变体，即在给定上下文的条件下，预测某个单词出现的概率，与 CBOW 模型不同的是，DM 模型中某个单词的上下文语境不仅是当前词汇的周围的 n 个词汇，还包括当前词汇所在的段落信息，该段落信息的引入是为了处理多义词的语境问题。DM 模型的结构如图 9 所示：

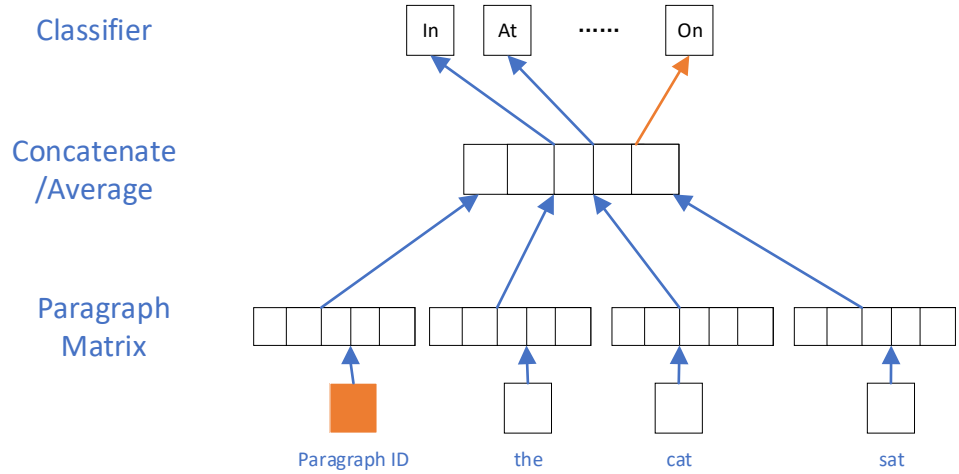


图 9 DM 模型结构

Figure 9 DM Model Structure

该模型是将文章段落以段落 ID 表示，此时段落 ID 是同词向量维度一样的向量，但两个向量的表示的向量空间是不同的，在映射层中，将两个两部分向量以拼接或者平均的方式连接起来，作为输出层的输入部分。在训练过程中，同一个段落或篇章中的词汇拥有共同的段落 ID，作为该词语的全局表示，相当于在训练模型时利用了整个句子的语义表达。同时，在预测阶段，对于目标段落，使用一个新的段落 ID 表示，词向量和输出层的参数不变，利用随机梯度下降法训练到误差损失收敛时，即可得到预测句子的向量表示。 Doc2Vec 的另一种模型为 Distributed Bags of Words（DBOW）模型，该模型与 Word2Vec 中的 Skip-Gram 模型类似，即都是在给定当前词的情况下预测该词的上下文语境。与 DM 模型同理，该模型在 Skip-gram 模型的基础上引入了段落 ID 来表示段落信息，该模型的结构如下所示：

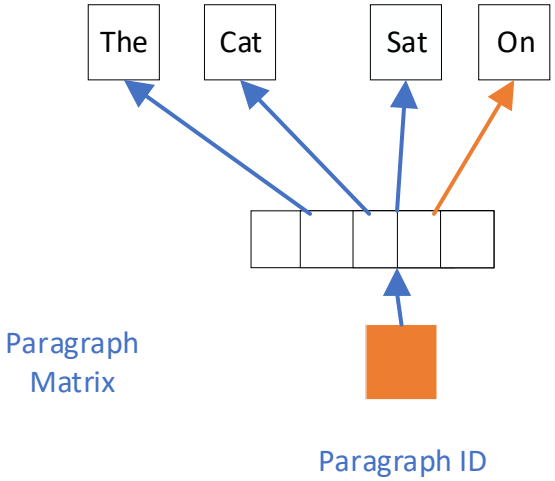


图 10 DM 模型结构

Figure 10 DM Model Structure

该模型是通过一个段落向量来预测某个随机词的概率分布，训练方式与 DM 模型一致，与 Skip-gram 模型相比，该模型降低了输入层的参数。

第四章 问题一

在处理网络问政平台的群众留言时，工作人员首先按照一定的划分体系对留言进行分类，以便后续将群众留言分派至相应的职能部门处理。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。针对这种现象，群众留言的高精度的分类算法就尤为重要。本题共有 9211 条留言数据，其主题分布如图 11 所示：

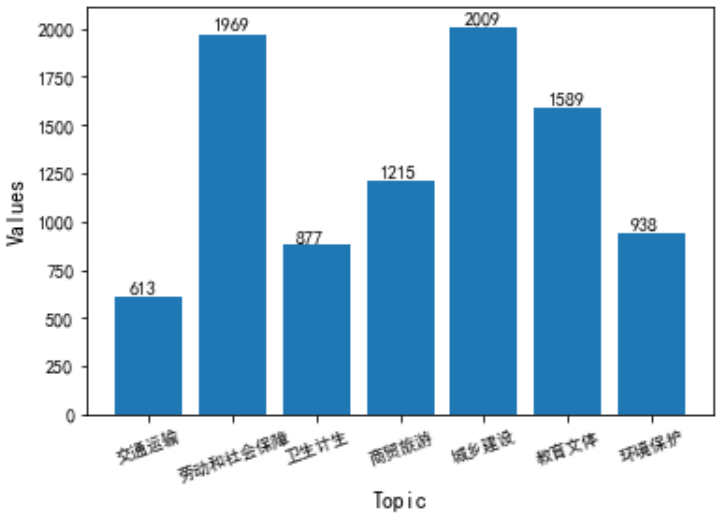


图 11 主题分布

Figure 11 Theme Distribution

由图 11 可以看出主题之间差异较大，例如主题“城乡建设”有 1600 个文本，而主题“交通运输”只有 613 个文本,训练数据集各类别不平衡。

4.1 文本分类机器学习算法

4.1.1 KNN

KNN 在 1951 年由 Fix 和 Hodge 在一份没有出版的美国空军学院的医学报告中最先被引入的，它是一种常见的非参数的模式识别的方法，该方法从 1960 年代开始被广泛应用于文本分类领域。KNN 是一种有效的简单分类算法，它被定义为监督学习算法。KNN 通过对待测样本和它周围相近的训练样本进行比较来学习。对待测样本的分类是基于它最近的邻居，通常这个邻居不止一个，所以叫做 K-NearestNeighbor(KNN)。这个 K 指的就是用来决定待测样本所属类别的 K 个邻居。KNN 算法有几个特点。第一个特点是：它的训练过程相对简单。KNN 对一个新的样本进行分类只需要该样本邻近的训练样本的类别，不需要通过所有的训练

样本来事先优化分类器参数。第二个特点是：只要两两文本之间的相似度或者距离被定义出来，它可以应用于用任何形式表达的文本分类。所以 KNN 可以看成是一个很“懒”的监督学习算法，不像其他的监督学习算法 KNN 在分类中只需要学习出待测样本最近的几个训练样本而不需要学习所有的样本。

KNN 分类器定义测试样本所属的类别是基于该样本周围最近的 k 个相邻的训练样本的类别。KNN 分类器通常通过欧几里德距离或者余弦相似度来计算两个样本之间的距离。通常 KNN 采用欧几里得距离。训练样本和测试样本之间的欧几里德距离描述如下：

X_i表示为有p个特征的输入样本$(x_{i1}, x_{i2}, \ldots, x_{ip})$，

n n 为所有输入样本的总数$(i = 1,2,\ldots,n)$

p为样本特征的总数$(j = 1,2,\ldots,p)$

样本 X_i 和样本$X_t(t = 1,2,\ldots,n)$之间的欧几里得距离定义为式(4.1)

$$d(x_i, x_t) = \sqrt{(x_{i1} - x_{t1})^2 + (x_{i2} - x_{t2})^2 + \cdots\ldots + (x_{ip} - x_{tp})^2} \tag{4.1}$$

写成一般的形式，样本$X_1 = (x_{11}, x_{12}, \ldots, x_{1n})$和样本$X_2 = (x_{21}, x_{22}, \ldots, x_{2n})$之间的欧几里得距离为

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2} \tag{4.2}$$

式(4.2)通过对样本X_1和X_2相对应的特征取差再取平方，把这些平方项累加之后再取平方根，最终得到样本X_1和X_2之间的欧几里得距离。为了防止不同的特征对应的量纲不同，通常在应用式(4.2)之前，需要对特征进行归一化处理。通常使用$min - max$归一化方法对样本的特征值进行归一化处理。假设V是样本A特征的特征值，通过式(4.3)将V处理成V'，V'的范围为$[0,1]$。

$$V' = \frac{V - A_{min}}{Amin_{max}} \tag{4.3}$$

其中A_{min}为该特征项的最小征值，A_{max}为特征A该特征项的最大特征值。通过对样本X最近邻的那些训练样本所属的类别分值大小排序，取最大值的类别作为样本X的类别

KNN 的优点是直观、易于实现和理解简单。但是 KNN 有一个很严重的缺点：当文本数量巨大的时候，一个待测样本无论何时进行分类，我们都需要对所有样本进行计算以得到该样本最近邻的那几个样本。这个阶段的计算是非常耗时的。KNN 算法的训练时间会和样本数以及样本的特征维度成正比。随着时间的推移，在文本信息系统中的文档数量会不断的增加，会不断的有大量的文档被增加或删除。如果使用 KNN 算法，很难实时地对增加的大量的文本做最近邻样本的计算。

4.1.2 SVM

支持向量机(Support Vector Machine,SVM)通过定义一系列的核函数把训练样本从一个非线性不可分的空间映射到一个线性可分的空间。从训练样本中可以产生两类样本的边界，这个边界是一个超平面。这两个超平面在该线性可分的空间中具有最大的间隔。而支持向量就是那些可以决定两个类别边界超平面的训练样本。SVM 在很多标准文本数据集上的分类性能都由于其他分类器。很多文本分类的研究中都把 SVM 作为分类器。SVM 最初是用来二值分类的，后续又出现了很多 SVM 的变种应用在多分类问题上。SVM 作为文本分类有好几个优点。第一，SVM 可以处理指数多的甚至无限多特征的样本，因为它不需要在转换过的样本空间中表示样本；第二，SVM 可以很好地处理像文本这样高维特征的样本，不需要对文本特征进行特征的降维。SVM 是目前为止分类性能最精确的分类算法之一，这点在各种不同领域的应用中都得到了证明。

假设训练数据集中样本 X，通过式(4.4)对该样本进行类别的划分：

$$f(X) = sign((X \cdot weight) + b) \tag{4.4}$$

其中$weight$和b是支持向量机的分类参数。$weight$是边界的超平面到分类线的距离向量，b是截距。$sign(\cdot)$是指示函数，如果输入大于 0，那么输出为正值；反之为负值。式(4.4)表示的式在二值分类中超平面的分类原则。输出要么是正值，要么是负值。$weight$通过训练样本的线性组合计算得到，如式(4.5)所示：

$$weight = \sum_{i=1}^{N} \alpha_i X_i \tag{4.5}$$

其中$\alpha_i (1 \le i \le N)$是作用在训练样本X_i上的拉格朗日乘子，N是训练样本的数量。结合式(4.5)，基于拉格朗日乘子的超平面方程可以写成式(4.5)：

$$f(X) = sign\left(\sum_{i=1}^{N} \alpha_i X \cdot X_i) + b\right) \tag{4.5}$$

支持向量机的基本思想是通过式(1.3)建立一个具有最大间隔的线性分类器。把原始空间映到一个线性可分的空间中，如图 12 所示：

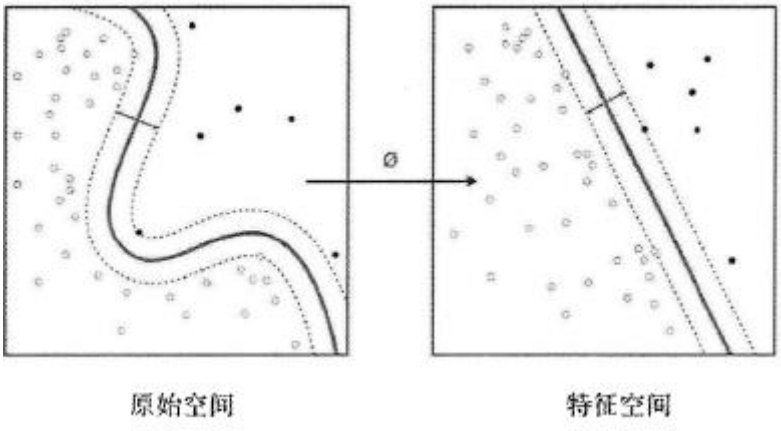


图 12 支持向量机特征空间映射

Figure 12 Support Vector Machine Feature Space Mapping

原始向量X映射到特征空间中声称新的特征向量$\psi(X)$,X和X_i生成了两个替换向量$\psi(X)$和$\psi(X_i)$，则式(4.5)改写成如下的形式：

$$f(X) = sign\left(\sum_{i=1}^{N} \alpha_i(\psi(X) \cdot \psi(X_i)) + b\right) \tag{4.6}$$

其中$\psi(X)$和$\psi(X_i)$的内积可以用X和X_i的内积的核函数来定义。使用核函数的目的是为了避免直接把X和X_i生成 $\psi(X)$和$\psi(X_i)$。因为$\psi(X)$和$\psi(X_i)$特征表达是困难的，有时甚至是无法表达的，而$\psi(X)$和$\psi(X_i)$的内积是容易求得的，而且通过核函数是非常容易实现的。通过使用核函数，式(4.6))改写成如下形式：

$$f(X) = sign\left(\sum_{i=1}^{N} \alpha_i(K(X \cdot X_i)) + b\right) \tag{4.7}$$

那么支持向量机的学习就转化成优化拉普拉斯乘子α_i的过程，α_i成为分类过程中唯一涉及到的参数。如果训练样本在投影到特征空间后是线性可分的，那么目标函数和它的限制条件如下所示：

$$\begin{cases} W(\alpha) = \sum_{i=1}^{N} \alpha_i - 0.5 \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j K(X_i X_j) y_i y_j \\ constraint: \alpha_i > 0 \end{cases} \tag{4.8}$$

其中y_i和y_j是训练样本X_i和X_j对应的不同的类别。如果训练样本在投影到特征空间后不是线性可分的，那么需要对式(4.7)进行最小化求解。加入松弛因子以允许一定程度上的样本错分。约束条件也做相应的修改，具体式(4.9)所示：

$$W(\alpha) = \sum_{i=1}^{N} \alpha_i - 0.5 \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j K\left(X_i \cdot X_j + \frac{\delta_{ij}}{C}\right) y_i y_j, \delta_{ij} = \begin{cases} 1 & if\ i = j \\ 0 & otherwise \end{cases} \tag{4.9}$$
$$constraint: \alpha_i > 0$$

其中δ是松弛变量，C是惩罚因子。

在训练支持向量机的过程中有很多优化拉格朗日乘子的策略。例如 chunking 方法、Qsuna 方法和 SMO 方法，其中 SMO 的优化速度是最快的也是最有效的。在 SMO 中，拉格朗日

乘子和b初始化为 0，通过两个训练样本解析优化其中一个样本对当前的拉普拉斯乘子进行更新，两个训练样本中一个训练样本是在式(4.9)中被错分的。这个更新过程一直进行到没有训练样本被错分为止。支持向量机的核心问题就是在给定的约束条件下优化拉普拉斯乘子来对目标函数求取最小值。

尽管如上所述，SVM 具有理论和实际优势，但它仅适用于二分类问题。如果用 SVM 来处理多分类问题，那么一个 SVM 是无法解决的需要多个 SVM 通过集成学习来处理，一个多分类问题需要分割成若干个二值分类问题。这样 SVM 就不适合当前的大数据量的文本分类和管理。特别在互联网时代的各种文本信息系统中属于不同的话题和类别的文本是在持续不断的增加，这都使得 SVM 在处理能力和处理效率上存在问题。

4.1.3 DT

决策树(Decisiontree,DT)是一种监督分类的学习方法。决策树的思想来自于由根和节点组成的树的结构，通过节点和连接节点的分支构成一个决策树。从根节点开始，向下移动从左至右生成中间节点和叶子节点。单个或更多分支可以从每个中间节点扩展生成。决策树可以线性化为决策规则，其中叶子节点的内容是最终决策的结果。决策树的主要优点是它解释简单，易于开发。此外，决策树可以方便地做增量而不需要重新训练所有的样本。决策树被很多的研究者作为文本分类器。在文本分类中，从根节点按照规则开始沿着路径向下，路径上的每个节点代表待测文本中包含的特征词,在这些节点产生的分支上面的值则是该节点所表示的特征词在测试文本中的权重,从根节点一直路径到叶子节点则代表了待测文本中包含的所有特征词，叶子节点则表示待测文本所属的类别。

在决策树中，通过节点和分支表征对象属性与对象值之间的一种映射关系。树中每个节点表示某个对象，而每个分叉路径则代表某个可能的属性值，而每个叶节点则对应从根节点到该叶节点所经历的路径所表示的对象的值。决策树最明显的特点之一是它的可理解性。通过决策树的路径很容易让人理解待测样本为什么被分类到一个特定的类别中。决策树是一个分层的判别分类的结构，在分类过程中当待测样本向下通过分支节点的时候，对分类判别不起作用的未知特征值会被删除。每一个分支都输出一个有关类别的分布。最终所有叶子节点的输出是所有分支的类别分布的总和累加在一起为 1。决策树通常在具有离散特征或者类别特征的样本分类上有更好的表现。就可理解性而言，这种符号分类器比“黑盒子”模型(如神经网络)具有优势。决策遵循的逻辑规则通过树状形式实现，这样要比神经网络中通过节

点之间大量的相互连接的权重调整参数来实现分类要简单的多。

4.1.4 RF

随机森林算法[17]是一种基于 Bagging 的集成学习方法,可以用于解决分类和回归问题,本文就随机森林算法用于分类问题进行研究.随机森林算法的基本构成单元是充分生长、没有剪枝的决策树.算法流程包括生成随机森林和进行决策两部分,如图 13 所示.

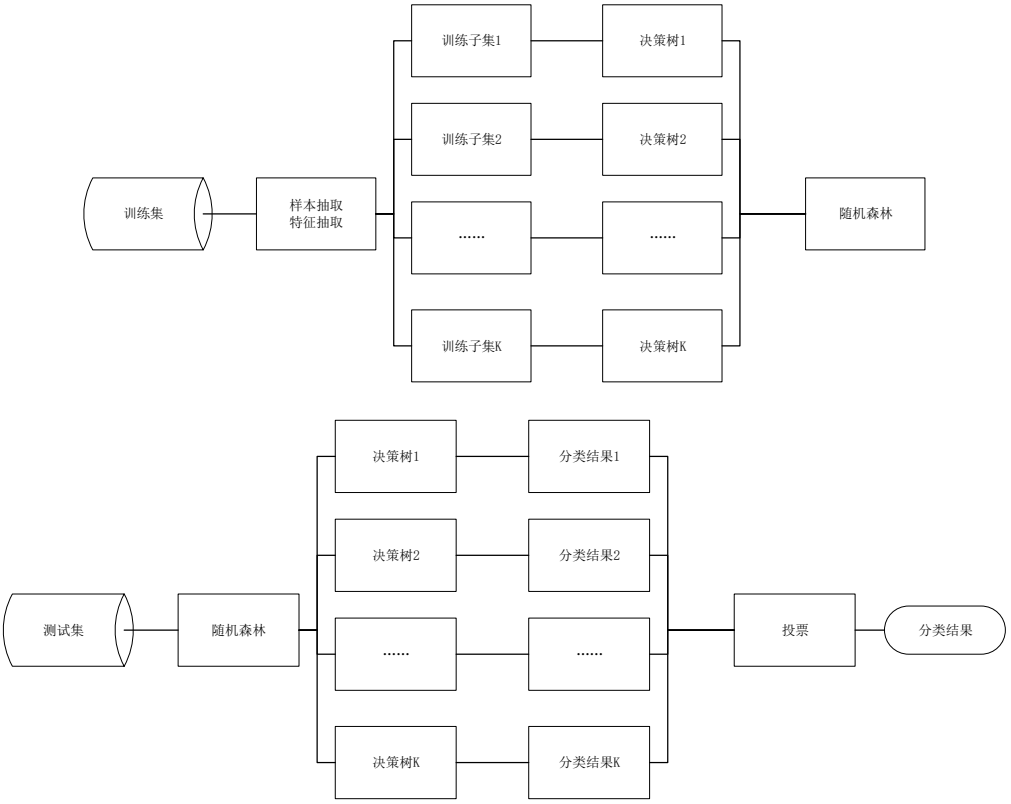


图 13 随机森林的算法流程

Figure 13 Algorithm Flow of Random Forest

具体的算法步骤如下:

Step1. 记给定原始训练集中的样本数量为 N ,特征属性数量为 M .采用 $bootsrap$ 抽样技术从原始训练集中抽取 N 个样本形成训练子集.

Step2. 从 M 个特征属性中随机选择 m 个特征作为候选特征 $(m \le M)$,在决策树的每个节点按照某种规则(基尼指数、信息增益率等)选择最优属性进行分裂,直到该节点的所有训练样例都属于同一类,过程中完全分裂不剪枝.

Step3. 重复上述两个步骤 k 次,构建 k 棵决策树,生成随机森林.

Step4. 使用随机森林进行决策,设x代表测试样本。h_i代表单棵决策树，Y代表输出变量即分类标签，I为指示性函数，H为随机森林模型,决策公式为:

$$H(x)=\arg\max_{Y}\sum_{i=1}^{k}I(h_i(x)=Y)$$

即汇总每棵决策树对测试样本的分类结果,得票数最多的类为最后的分类结果.

4.1.5 XGBoost

XGBoost 算法[18]是一种基于 Boosting 的集成学习算法。它在 GBDT 算法的基础上进行了改进。 XGBoost 算法不仅提高了速度，而且还提高了准确性，从而对成本函数进行了二阶泰勒展开。引入正则化术语以避免过度拟合。集成树模型为：

$$\hat{y}_i=\theta(x_i)=\sum_{k=1}^{K}f_k(x_i)\ \ f_k\in F \tag{4.10}$$

其中，K是子模型的总数；$F=\{f(x)=\omega_{q(x)}\}$是所有回归树的集合，$\omega_{q(x)}$是由回归树的所有叶节点的权重组成的权重向量；$\hat{y}_i$是样本预测值；x_i是样本输入功能；f_k是第 k 个回归树，每个回归树具有独立的叶权重ω和树结构q。如等式(4.11)所示引入目标函数。

$$O=l(y_i,\hat{y}_i)+\sum_{k=1}^{K}\Omega(f_k) \tag{4.11}$$

其中，$l(y_i,\hat{y}_i)$是损失函数，代表预测值$\hat{y}_i$与实际值y_i之差；Ω是正则化项，用于平滑最终学习权重以避免过度拟合。通过加法进行多次迭代，定义$\hat{y}_i^{(t)}$为第 i 次迭代中样本 i 的预测值，合成损失函数和正则项，并结合公式(4.10)和公式(4.11)表示为：

$$O^{(t)}=\sum_{i=1}^{N}l(y_i,\hat{y}_i^{(t-1)}+g_i f_t(x_i)+\frac{1}{2}h_i f_t^{\,2}(x_i))+\Omega(f_t) \tag{4.12}$$

$$g_i=\partial_{\hat{y}^{(t-1)}}l(y_i,\hat{y}^{(t-1)}),h_i=\partial^2_{\hat{y}^{(t-1)}}l(y_i,\hat{y}^{(t-1)})$$

其中，N是树木的数量。主要实现过程如下：

Step1. 建立新的决策树；

Step2. 根据等式(4.12)所示的目标函数计算出每个训练样本的g_i和h_i，并开始迭代；

Step3. 使用近似贪婪算法找到最佳分割点，以获得决策树结构$f_t(x)$，其中t表示第t

次迭代；

Step4. 添加 $f_t(x)$ 到集成树模型中；

Step5. 按照步骤(1)-(4)执行多次迭代以获得最终分类模型。

4.2 文本分类深度学习算法

4.2.1 TextCNN

卷积神经网络是一类包含卷积计算且具有深度结构的前馈神经，是深度学习(Deep Learning)的代表算法之一。卷积神经网络首次应用于文本分类是在 2014 年纽约大学 Yoon Kim 提出的 TextCNN 模型，对于文本的处理过程如图 1 所示。为不同尺寸的卷积核都建立一个卷积层。所以会有多个 Feature Map。将得到的多个特征进行融合后，连接到池化层。最大池化层只会输出最大值，对输入中的补 0 做过滤。全连接层的每一个结点都与上一层的所有结点相连，用来把前边提取到的特征综合起来，每个神经元的激励函数一般采用 ReLU 函数。输出层的输入为全连接层的输出，经过 Softmax 层作为输出层。对于多分类问题可以使用 Softmax 层，输出每个类别的概率。

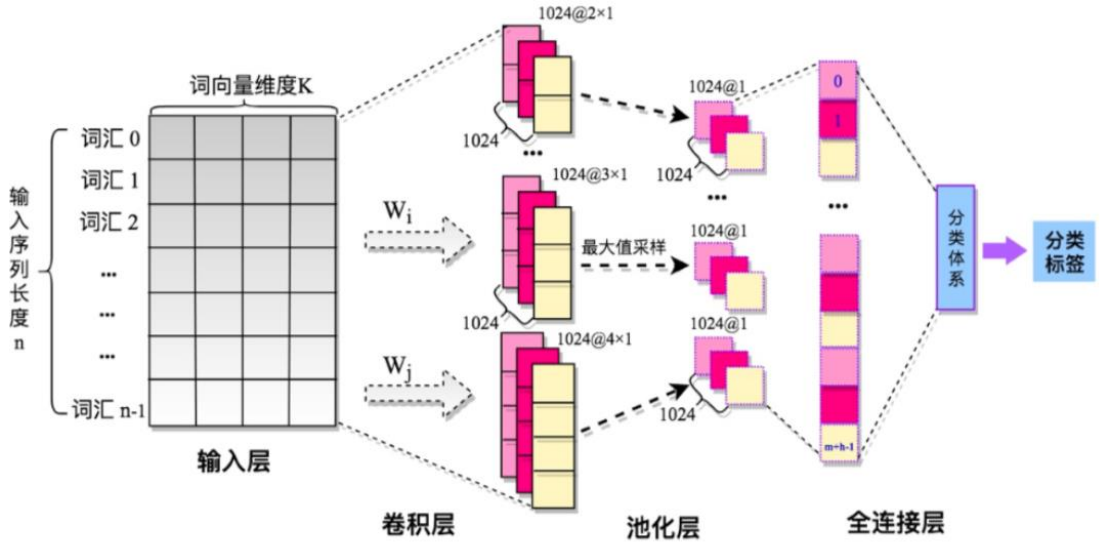


图 14 文本处理过程

Figure 14 Text Processing

4.2.2 BiLSTM

LSTM 是一种循环神经网络 RNN(Recurrent Neural Networks)，模型如图 4.5 所示，它

的改进是在 RNN 的基础上加入“门控”来处理信息的传递。符合规则的信心会被留下，不符合规则的信息会被遗忘。这样有效地解决了 RNN 的梯度消失或梯度爆炸问题，LSTM 的优点是可以捕获序列数据中的长期依赖关系，适用于处理时间序列高度相关的问题。

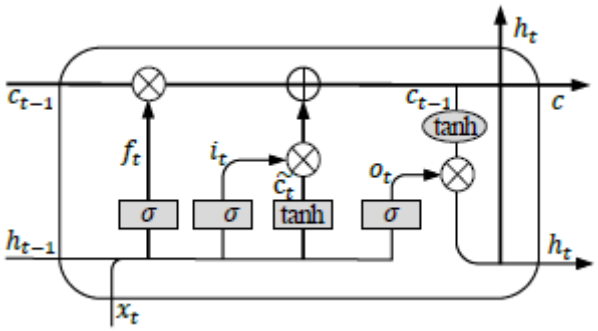


图 15 LSTM 模型结构

Figure 15 LSTM Model Structure

LSTM 模型由一系列相同的模块单元连接而成，每个模块包括：遗忘门f_t、输入门i_t、输出门o_t和记忆单元C_t。LSTM 中重要公式如下：

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

$$i_t = \sigma[w_i \cdot (h_{i-1}, x_i) + b_i]$$

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

$$\tilde{C}_i = \tanh[w_c \cdot (h_{i-1}, x_i) + b_c]$$

$$o_t = \sigma[w_o \cdot (h_{t-1}, x_t) + b_o]$$

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

其中，w_f、w_i、w_o、w_c是权重矩阵，b_f、b_i、b_c、b_o为 LSTM 模型的偏移量，h_t为t时刻的隐藏状态。

双向 LSTM(bi-direction LSTM，BiLSTM)是在 LSTM 的基础上，借鉴人类理解文的前后联系思想，引入正负时间方向概念形成的变形结构。BiLSTM 是将时序相反的两个 LSTM 网络连接到同一输出，运用前向隐含层结点和后向隐含层结点，分别捕获上下文信息。两个隐含层的结果共同作用，输出最终结果。

BiLSTM 神经网络结构如图 16 所示，每个隐层单元保存两个信息：A和A^*，A参与正向运算，A^*参与反向运算，二者共同决定最终的输出值y。当进行正向运算时，隐层单元S_t和S_{t-1}相关；反向运算时，隐层单元S_t^*和S_{t+1}^*相关。

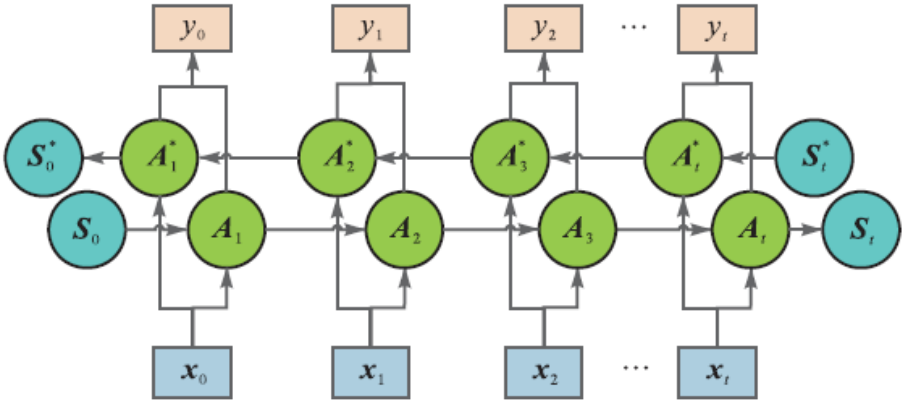


图 16 BiLSTM 神经网络结构

Figure 16 BiLSTM Neural Network Structure

对于不依赖因果关系的时间序列，BiLSTM 利用已知时间序列和反向位置序列通过正反双向运算，加深对原序列特征提取层次，提高模型输出结果的准确性。因此 BiLSTM 在解决时序性问题方面往往能够取得比单向 LSTM 更好的效果。

相比于其他神经网络结构，LSTM 更加真实地模拟了人类的行为逻辑和神经认知过程，尽管目前在一些复杂任务中，以 CNN 为代表的前馈神经网络更具性能优势，但 LSTM 网络处理深层次复杂任务的潜力更加值得期待。

4.2.3 BiLSTM_Attention

注意力(Attention)机制是指在文本句中每个词或字符对句子语义的贡献程度都是不同的，它能从众多信息中选出对当前任务最关键的信息。

首先将 BiLSTM 隐藏层中的输出h_t经过一层非线性转换得到h，如下式所示：

$$\tilde{h}_t = \tanh\left(w_w h_t + b_w\right)$$

w_w和b_w为注意力计算过程中的权重矩阵和偏置项，w_w随机初始化，随模型训练不断更新，然后通过一层 Softmax 进行归一化操作得到注意力权重矩阵α_t，如下式所示：

$$\alpha_t = softmax(\tilde{h}_t)$$

将α_t和h_t进行操作得到经过注意力机制的最终向量h'_t，如下式所示：

$$h'_t = \sum_i \alpha_i \, h_t$$

最后在 Attention 后面连接一个 softmax 输出预测值。

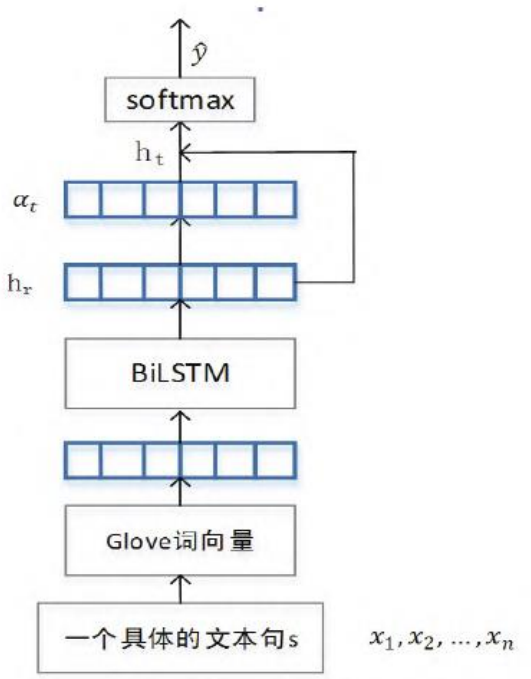


图 17 BiLSTM_Attention 模型结构

Figure 17 BiLSTM_Attention Model Structure

4.2.4 TextRCNN

CNN 类模型可以获得局部特征，如关键词与关键短语的提取等语义特征，但是 CNN 类的模型往往需要手动确定卷积核的尺寸。太小的尺寸可能丢失一些十分重要的信息，而太大的尺寸会使待学习的参数规模大大增加，加大了模型训练的难度。为了解决这一问题，TextRCNN 模型被提出，在此模型中，首先使用双向 LSTM 网络处理输入向量，然后在每个时间步，把循环网络的输出与对应的词向量拼接，作为当前时间步的“语义向量”。语义向量可以很好地掌握文本的上下文特征。最后使用了纵向最大池化进行特征选择，选取最重要的语义向量作为输出的特征。通过池化层与循环网络的连接，TextRCNN 结合了卷积网络与循环网络的优点，既可以获取局部特征也可以获取上下文特征。

对长度为n的文本

$$x_{1:n} = x_1 \oplus x_2 \oplus x_3 \cdots \oplus x_n,$$

令$c_l(x_i)$代表第i个词的左语义向量，$c_r(x_i)$代表右语义向量，$e(x_i)$代表第i个词的词向量。

$$c_l(x_i) = f(w^l \cdot c_l(x_{i-1}) + W^{sl} \cdot e(x_{i-1})),$$

$$c_r(x_i) = f(w^r \cdot c_r(x_{i-1}) + W^{sr} \cdot e(x_{i+1})),$$

把词向量和对应的左右语义向量拼接，得到词的语义向量

$$t_i = \tanh(W[c_l(x_i); e(x_i); c_r(x_i)] + b),$$

之后，对所有的语义向量，进行纵向的最大池化操作：

$$t = \max_i(t_i)$$

得到特征后输出到分类器即可。

图 18 是 TextRCNN 的典型结构，经过嵌入层得到的词向量与正逆向长短是循环网络拼接并使用一个反曲正弦函数激活得到，之后把文章所有的语义向量取纵向最大值得到特征，这里也可以使用最大的 k 个向量作为特征，之后把特征输入到分类器即可。

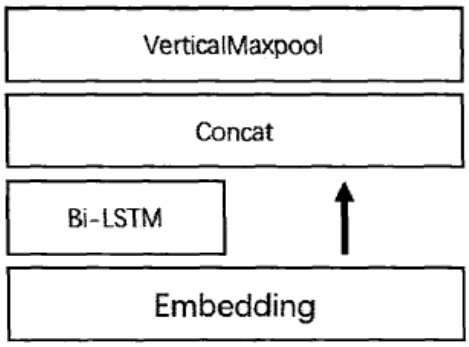


图 18 TextRCNN 结构

Figure 18 TextRCNN structure

4.2.5 DPCNN

DPCNN 是由 Textcnn 发展而来，利用 CNN 来实现文本分类中的特征表达，是文本分类领域词层面的深度学习模型，因网络的序列长度逐层减少而形似金字塔，从而得名。DPCNN 能够学习更复杂的非线性特征，提取更深层次特征，从而高效地建立长距离文本的语义。图 2 为 DPCNN 结构，在叠加模块中，除了卷积层外，网络包含步长为 2 的下采样，使得模型能够在文本中有效地表征词之间的长程关联性，保留更多的全局信息；同时使计算量逐层大幅减少，极大地提升训练和预测的运行效率。此外，网络还包含一个残差连接，这样有利于深层网络的训练。

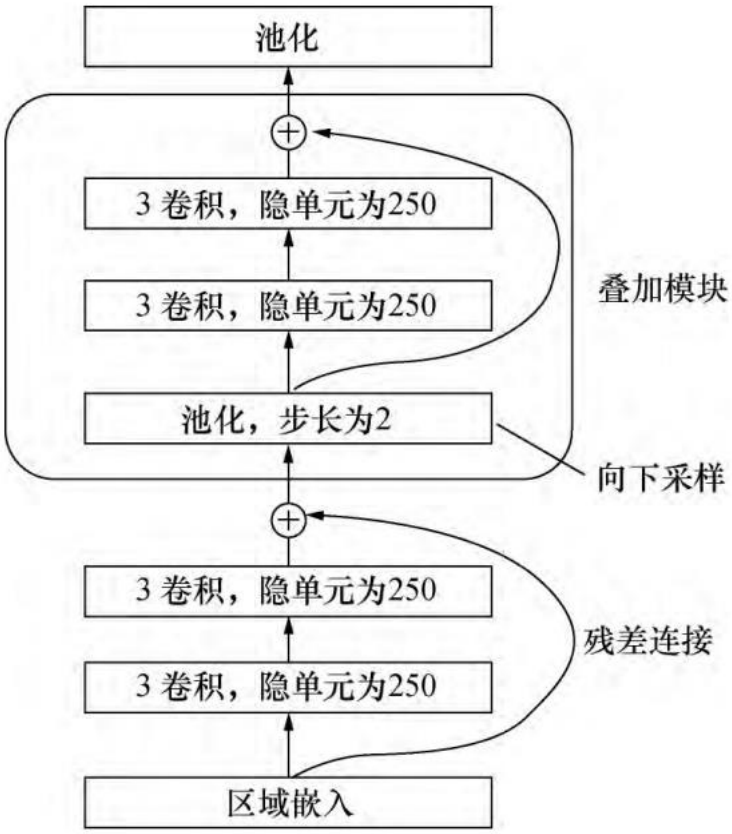


图 19 深度金字塔卷积神经网络

Figure 19 DPCNN structure

4.2.6 Transformer

Transformer 模型由 6 个编码器层和 6 个解码器层堆叠组成。待校对文本在编码器端输入，正确文本在解码器输入，通过监督学习来训练模型。即训练阶段讲训练集中的输入语料作为编码器输入，讲对应的正确语料作为编码器的输入；在测试阶段将测试集的待校对语料作为编码器的输入，解码器端无输入，仅依赖前一时刻的解码器输出信息进行校对。器整体结构如图所示。

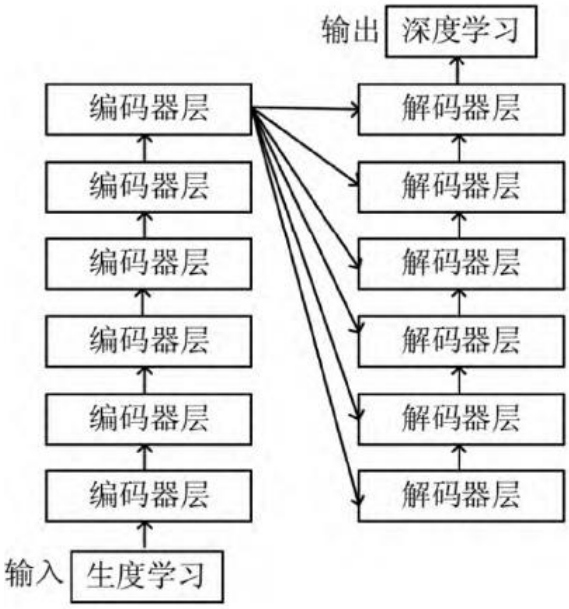


图 20 Transformer 模型整体结构

Figure 20 Overall structure diagram of Transformer model

编码器层内部包含两个子层，即一个 Self-Attention 层和一个全连接的前馈神经网络层。Self-Attention 层中的查询向量、键向量和值向量均来自于前一个编码器层，因此编码器层的每个位置都能去关注前一层输出的所有信息，使得当前节点不仅关注当前信息，还能获得上下文的语义信息。前馈神经网络层应用于 Self-Attention 层的输出，由两个线性变换和一个 ReLU 激活函数组成。计算方法如式(4.13)所示。

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{4.13}$$

其中，W_1和W_2为模型中神经元的权值，b_1和b_2为偏置值，x为输入向量。

编码器层与解码器层的结构如图 21 所示。

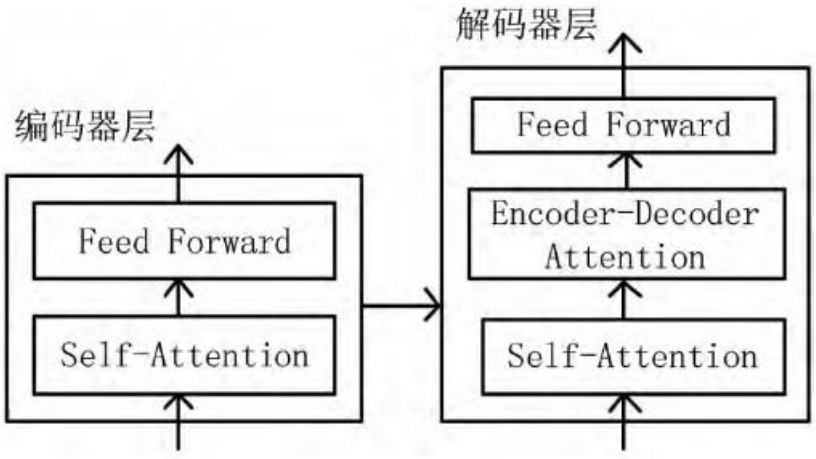


图 21 编码器层与解码器层内部结构图

Figure 21 Internal structure diagram of encoder layer and decoder layer

解码器层不仅包含编码器层中的两个子层，还添加了一个注意力子层对编码器的输出执行多头注意，其查询向量来自于前一个解码器层，键向量和值向量来自于编码器的输出，因此解码器的每个位置都可以关注到编码器输入序列的全部信息，帮助当前解码器节点获取到需要关注的重点内容。此外，解码器的 Self-Attention 子层加入了 masked 部分，其可以对某些值进行掩盖，从而防止模型注意到后续位置信息。这种屏蔽确保了当前的预测只能依赖于之前的已知输出。

Transformer 模型的内部结构如图 22 所示。

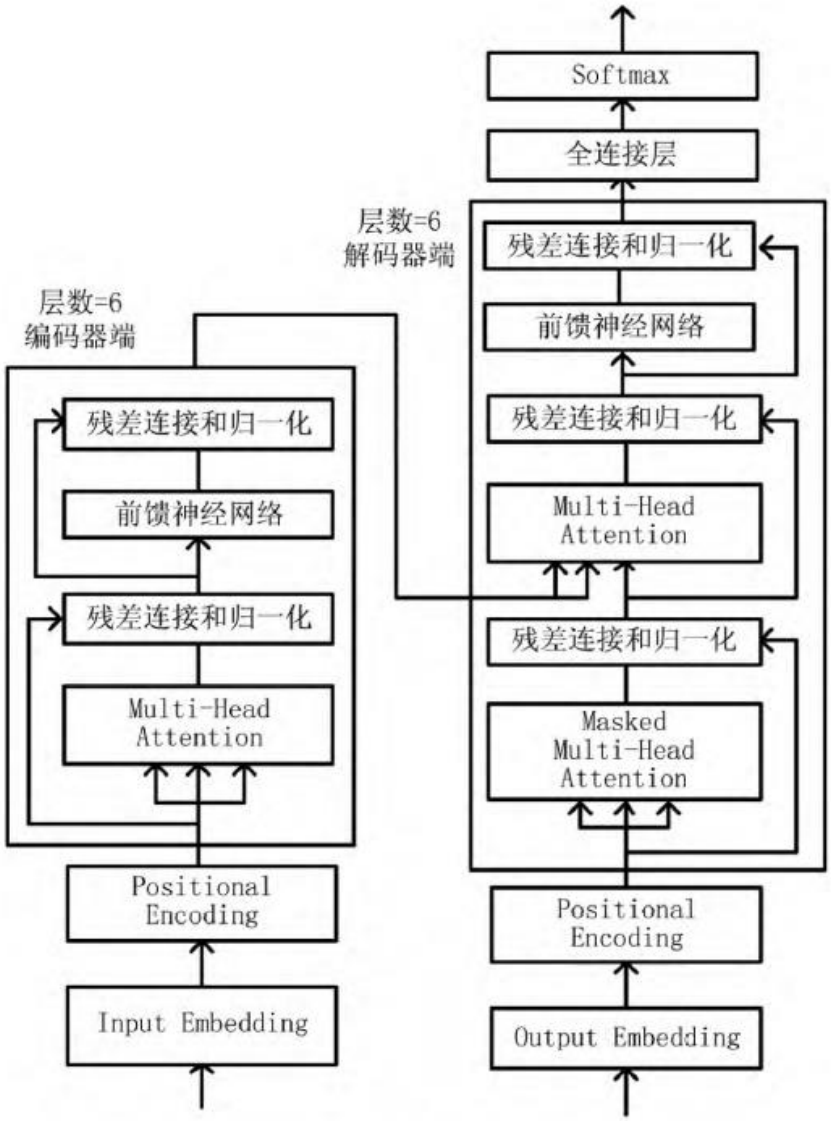


图 22 Transformer 模型的内部结构

Figure 22 The internal structure of the Transformer model

- Embedding

首先，模型对输入数据进行 Embedding，即词嵌入，将输入词语转变成向量。将向量输

入到编码器和解码器的第一层，经过多头注意力机制处理后传入前馈神经网络，得到的输出信息作为下一层编码器和解码器的输入。

- Positional Encoding

因为 Transformer 模型缺少对输入序列中词语顺序的表示，所以需要在编码器层和解码器层的输入添加一个 Positional Encoding 向量，即位置编码向量，维度与输入向量的维度相同,该向量决定当前词在序列中的位置，计算方法入式 4.14，4.15 所示

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{model}}\right) \tag{4.14}$$

$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{model}}\right) \tag{4.15}$$

其中，pos是指当前词语在句子中的位置；i表示pos对应的向量值，取值范围为 0~255；d_{model}表示向量维度。在偶数位置，使用正弦编码；在奇数位置，使用余弦编码。最后将位置编码向量与输入向量相加，作为输入传入模型。

- 残差连接和归一化

编码器层和解码器层中每个子层都加入了残差连接和归一化。子层先进行残差连接，避免误差反向传播时的梯度消失，接着对输出进行归一化，避免梯度消失或梯度爆炸。剩余连接和归一化后的输出表示如式 4.16 所示：

$$SubLayer_{output} = LayerNorm(x + (SubLayer(x))) \tag{4.16}$$

其中，x为前一层前馈神经网络或多头注意力层的输出向量，$SubLayer$为注意力机制函数，$SubNorm$为归一化函数。

- 输出层

当解码器层全部执行完毕后，为了将得到的向量映射为本文需要的词语，需要在最后一个解码器层后添加一个全连接层和$Softmax$层。全连接层输出$logits$向量，作为$Softmax$层的输入。假设词典包括n个词语，那最终$Softmax$层会输出n个词语分别对应的概率值，概率值最大的对应词语就是最终的输出结果。

4.3 基于机器学习算法的群众留言分类实验分析

- 模型构建

针对本题所给的群众留言文本数据，分别采用其中的留言标题数据，留言详情数据进行建模。对不同的文本特征提取方法，不同的分类算法进行组合实验。基于机器学习算法的群众留言分类模型如图所示：

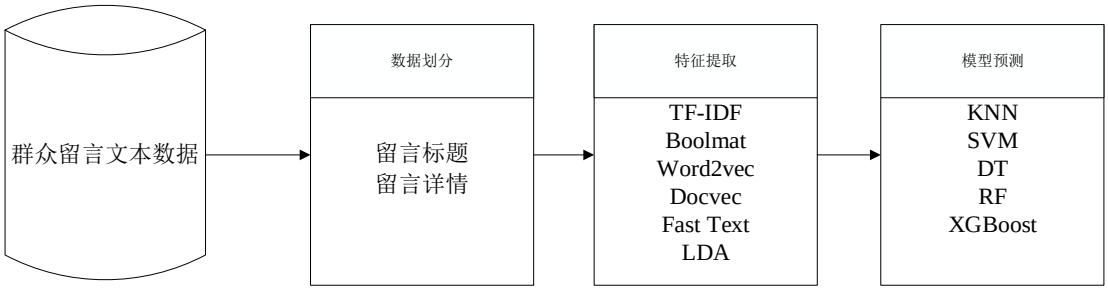


图 23 基于机器学习算法的群众留言分类模型

Figure 23 Crowd Message Classification Model based on Machine Learning Algorithm

- **实验与分析**

为了验证各个特征提取方法与分类模型组合的有效性，分别基于留言标题数据和留言详情数据进行实验，将数据集 1:1 划分为训练集和测试集。采用 3 项指标评估算法优劣。ACC 表示分类正确率，macro F1 表示 F1 值宏平均，micro F1 表示 F1 值微平均，它们是分类任务中最常用的性能指标，值越高代表分类越准确。各算法对比结果如表 3，表 4 所示。

表 3 基于留言标题数据的分类算法对比

Table 3 Comparison of Classification Algorithm based on Message Title Data

算法	macro F1	micro F1	ACC
Boolmat-XGBoost	0.814292225	0.810423453	0.810423453
Boolmat-KNN	0.599472838	0.623669924	0.623669924
Boolmat-SVM	0.838136016	0.840173724	0.840173724
Boolmat-RF	0.828936745	0.832790445	0.832790445
Boolmat-DT	0.74133399	0.749402823	0.749402823
TF-IDF-XGBoost	0.715329486	0.697502714	0.697502714
TF-IDF-KNN	0.745090155	0.755266015	0.755266015
TF-IDF-SVM	0.818559323	0.819543974	0.819543974
TF-IDF-RF	0.782112838	0.776764387	0.776764387
TF-IDF-DT	0.720562997	0.718566775	0.718566775
LDA-XGBoost	0.494021396	0.513789359	0.513789359
LDA-KNN	0.350216854	0.387187839	0.387187839
LDA-SVM	0.367908931	0.424538545	0.424538545
LDA-RF	0.466383874	0.48990228	0.48990228
LDA-DT	0.400790671	0.417155266	0.417155266
Word2vec(dim=100)-XGBoost	0.712947	0.730945	0.730944625
Word2vec(dim=100)-KNN	0.755769	0.76873	0.768729642
Word2vec(dim=100)-SVM	0.748112	0.760694897	0.760694897
Word2vec(dim=100)-RF	0.713618	0.743974	0.743973941
Word2vec(dim=100)-DT	0.424191	0.454723	0.454723127

Word2vec(dim=300)-XGBoost	0.731599502	0.747014115	0.747014115
Word2vec(dim=300)-KNN	0.743331032	0.754831705	0.754831705
Word2vec(dim=300)-SVM	0.770661648	0.776112921	0.776112921
Word2vec(dim=300)-RF	0.69833573	0.733984799	0.733984799
Word2vec(dim=300)-DT	0.402792601	0.429098806	0.429098806
FastText(dim=100)-XGBoost	0.750685034	0.765906623	0.765906623
FastText(dim=100)-KNN	0.717968126	0.735767058	0.735767058
FastText(dim=100)-SVM	0.738545724	0.753368101	0.753368101
FastText(dim=100)-RF	0.733714351	0.755758366	0.755758366
FastText(dim=100)-DT	0.471366502	0.49934811	0.49934811
FastText(dim=200)-XGBoost	0.763843004	0.777270752	0.777270752
FastText(dim=200)-KNN	0.712638583	0.729682747	0.729682747
FastText(dim=200)-SVM	0.748596844	0.761842677	0.761842677
FastText(dim=200)-RF	0.740933735	0.761190787	0.761190787
FastText(dim=200)-DT	0.510558017	0.541938288	0.541938288
Docvec(dim=100)-XGBoost	0.7449976	0.758957655	0.758957655
Docvec(dim=100)-KNN	0.718160464	0.733116178	0.733116178
Docvec(dim=100)-SVM	0.746212683	0.760043431	0.760043431
Docvec(dim=100)-RF	0.752495138	0.768078176	0.768078176
Docvec(dim=100)-DT	0.513900757	0.535722041	0.535722041
Docvec(dim=300)-XGBoost	0.760741375	0.773724213	0.773724213
Docvec(dim=300)-KNN	0.712879124	0.727035831	0.727035831
Docvec(dim=300)-SVM	0.763358866	0.772204126	0.772204126
Docvec(dim=300)-RF	0.753505828	0.770249729	0.770249729
Docvec(dim=300)-DT	0.528429362	0.557437568	0.557437568

表 4 基于留言标题数据的分类算法对比

Table 4 Comparison of Classification Algorithm based on Message Title Data

算法	macro F1	micro F1	ACC
LDA-XGBoost	0.652239847	0.697937025	0.697937025
LDA-KNN	0.569890613	0.627578719	0.627578719
LDA-SVM	0.477445151	0.571769815	0.571769815
LDA-RF	0.639343676	0.692725299	0.692725299
LDA-DT	0.518128628	0.555266015	0.555266015
TF-IDF-XGBoost	0.81327443	0.819978284	0.819978284
TF-IDF-KNN	0.816291802	0.823235613	0.823235613
TF-IDF-SVM	0.870497463	0.877741585	0.877741585
TF-IDF-RF	0.819466173	0.830401737	0.830401737
TF-IDF-DT	0.706995751	0.717046688	0.717046688
Word2vec(dim=100)-XGBoost	0.801569972	0.823452769	0.823452769
Word2vec(dim=100)-KNN	0.79658984	0.819109663	0.819109663
Word2vec(dim=100)-SVM	0.84104519	0.859066232	0.859066232
Word2vec(dim=100)-RF	0.795335272	0.826275787	0.826275787
Word2vec(dim=100)-DT	0.608235076	0.637350706	0.637350706
Word2vec(dim=300)-XGBoost	0.794972321	0.817806732	0.817806732
Word2vec(dim=300)-KNN	0.796355639	0.814766558	0.814766558
Word2vec(dim=300)-SVM	0.854183673	0.868403909	0.868403909
Word2vec(dim=300)-RF	0.770108055	0.807383279	0.807383279
Word2vec(dim=300)-DT	0.566337583	0.597176982	0.597176982
FastText(dim=100)-XGBoost	0.828718828	0.851900109	0.851900109
FastText(dim=100)-KNN	0.810389239	0.829750271	0.829750271
FastText(dim=100)-SVM	0.839223607	0.856677524	0.856677524
FastText(dim=100)-RF	0.801104462	0.831704669	0.831704669
FastText(dim=100)-DT	0.608493389	0.642996743	0.642996743
FastText(dim=150)-	0.834402595	0.853420195	0.853420195

XGBoost			
FastText(dim=150)-KNN	0.810854042	0.829967427	0.829967427
FastText(dim=150)-SVM	0.846790804	0.863409338	0.863409338
FastText(dim=150)-RF	0.805440395	0.835396308	0.835396308
FastText(dim=150)-DT	0.597442916	0.633224756	0.633224756
Docvec(dim=100)-XGBoost	0.786162908	0.815635179	0.815635179
Docvec(dim=100)-KNN	0.785357405	0.809554832	0.809554832
Docvec(dim=100)-SVM	0.82257431	0.842996743	0.842996743
Docvec(dim=100)-RF	0.773145145	0.815200869	0.815200869
Docvec(dim=100)-DT	0.562946972	0.596959826	0.596959826
Docvec(dim=300)-XGBoost	0.802263073	0.826927253	0.826927253
Docvec(dim=300)-KNN	0.794998232	0.813897937	0.813897937
Docvec(dim=300)-SVM	0.854301898	0.864929425	0.864929425
Docvec(dim=300)-RF	0.761003094	0.810206298	0.810206298
Docvec(dim=300)-DT	0.51047591	0.544625407	0.544625407
Docvec(dim=300)-DT	0.528429362	0.557437568	0.557437568

根据实验分析，采用留言详情数据进行建模分类的三项指标都高于采用留言主题数据的。通过不同方法组合的实验，发现 FastText，TF-IDF 方法在本问题中 F1 值与 ACC 都高于其他方法；在分类算法中 SVM 和 XGBoost 方法在保持高精度的情况下，稳定性也保持良好。其中 TF-IDF-SVM 的 macro F1 为 **0.870497463**，ACC 为 **0.877741585** 远高于其他算法，FastText(dim=150)-XGBoost 的 macro F1 为 **0.834402595**，ACC 为 **0.853420195** 也表现出了准确度上的优势

4.4 基于深度学习算法的群众留言分类实验分析

- **模型构建**

针对本题所给的群众留言文本数据，采用其中的留言标题和留言详情数据合并后进行建模，对不同的神经网络算法进行实验。基于机器学习算法的群众留言分类模型如图所示：

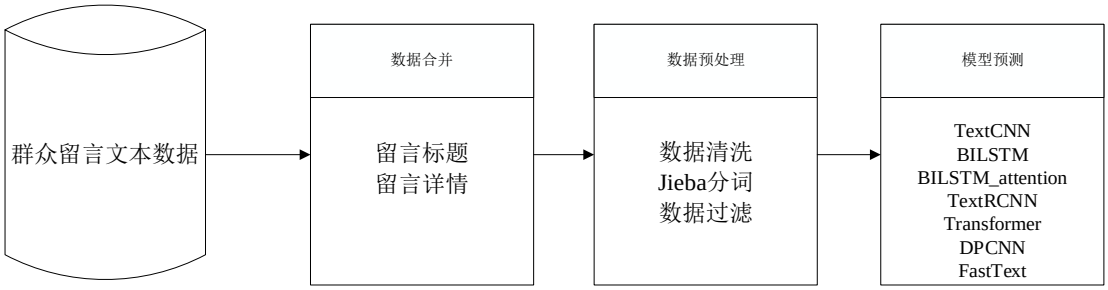


图 24 基于深度学习算法的群众留言分类模型

Figure 24 Classification Model of Public Comments based on Deep Learning Algorithm

- **实验与分析**

为了验证各个深度学习模型的有效性,基于留言标题数据和留言详情数据的合并数据进行实验，将数据集 8:1:1 划分为训练集,测试集,验证集。采用 3 项指标评估算法优劣。ACC 表示分类正确率，macro F1 表示 F1 值宏平均，micro F1 表示 F1 值微平均，它们是分类任务中最常用的性能指标，值越高代表分类越准确。各算法对比结果如表 5 所示。

表 5 基于深度学习算法对比结果

Table 5 Compare the Results based on Deep Learning Algorithm

算法	macro F1	micro F1	ACC
TextCNN epoch=20 batch size=32	0.875026195	0.883821933	0.883821933
TextCNN epoch=20 batch size=128	**0.900545333**	**0.903365907**	**0.903365907**
DPCNN epoch=20 batch size=128	0.895310849	0.901194354	0.901194354
Transformer epoch=10 batch size=128	0.701054804	0.736156352	0.736156352
FastText epoch=20 batch size=128	0.885583369	0.891422367	0.891422367
TextRNN epoch=2 batch size=128	0.074545928	0.208469055	0.208469055
TextRNN_Att epoch=2 batch size=128	0.052844501	0.226927253	0.226927253

在实验中发现由于算力不够，例如 BILSTM,TextRCNN，结构复杂的深度学习模型，无法在 PC 端训练完成。根据表 2 的实验发现 batch_size 为 128 时，神经网络模型的三项指标表现都有提升。在低算力的情况下，TextCNN,DPCNN,FastText 群众留言分类上的训练时间上优于结构复杂的神经网络模型。其中 TextCNN 的 macro F1 达到了 0.900545333，DPCNN 的 macro F1 达到了 0.895310849。与基于传统机器学习方法,在 macro F1 上最低提升了 **4.022%**，

最高提升了 47.358%；在 ACC 上低提升了 2.837%，最高提升了 39.712%。在分类精度上，深度学习算法具有优势性。

4.5 基于进化算法的机器学习算法群众留言分类优化

4.5.1 标准 PSO 算法

PSO 算法的运行过程为：算法初始时期在 D 维搜索空间中首先随机初始化 n 个粒子，经过 k 次迭代，粒子 j 的位置$(x_{j1}^k, x_{j2}^k, ..., x_{jD}^k)$和速度$(v_{j1}^k, v_{j2}^k, ..., v_{jD}^k)$，以一定的速度在搜索空间中移动，粒子 j 依据当前的速度追踪个体极值 p_{lbest} 和群体极值 p_{gbest}，然后进行种群的更新。在每一次迭代中，根据式(1),(2)粒子 j 的速度和位置可更新为：

$$v_{jd}^{k+1} = wv_{jd}^k + c_1 rand(p_{lbest}^k - x_{jd}) + c_2 Rand(p_{kbest}^k - x_{jd}) \tag{4.17}$$

$$x_{jd}^{k+1} = x_{jd}^k + v_{jd}^k \tag{4.18}$$

其中，为第 j 个粒子第 k 次迭代时在第 d 维搜索空间中的速度，ω 为惯性权重，c_1，c_2 为加速因子，rand 和 Rand 为[0,1]，p_{lbest} 为第 j 个粒子在第 k 次迭代时历史最优位置，p_{gbest} 为第 k 次迭代时群体的历史最优位置。

4.5.2 改进的 CPSO 算法

PSO 算法中，惯性权重的设置决定了粒子的探索和搜索能力，对算法性能的发挥起到至关重要的作用。通过对 w 的调整可以调节粒子全局搜索和局部搜索的能力的大小，达到二者平衡。若 w 较大，则粒子的全局搜索能力强，反之，粒子的局部搜索能力强。惯性权重的设置通常采取随着迭代次数的增加在某个区间内线性递减的策略，计算公式如下：

$$w = w_{max} - (w_{max} - w_{\min}) \times \frac{k}{maxgen} \tag{4.19}$$

其中 w_{max} 和 $w_{\min}$ 分别为惯性权重的上限和下限，k 为当前迭代次数，$maxgen$ 为最大迭代次数。

这种线性递减的方法在初期阶段的值比较大，此时粒子的全局搜索能力比较强，而局部搜索能力较弱，有利于快速定位最优解的大致位置。然而随着迭代次数的增加，w 的值逐渐减小，后期种群收敛速度较慢，粒子局部搜索的能力较强，而全局搜索的能力较弱，由于粒

子具有只向自身最好飞行位置和种群中最好粒子的飞行位置学习的特性，使得算法容易出现早熟现象。而且单纯地依靠线性变换，粒子的权重无法依靠种群的多样性精确地调节[19]。

其次，当算法陷入局部极值时，种群粒子在局部极值位置附近聚集并重复类似的寻优轨迹，导致算法很难摆脱局部极值[20]。混沌是一种非线性的自然现象，具有随机性、遍历性等特点，可进行寻优搜索[21][22]，本文借鉴混沌现象随机性和遍历性的特点，提出了基于混沌优化摆脱局部极值的方法。

一个常用的混沌模型 Logistic 方程如下：

$$z_{n+1} = \mu z_n(1-z_n), n = 0,1,2,... \tag{4.20}$$

其中，μ 为控制参量，当 0≤z0≤1，μ=4 时，Logistic 处于完全混沌状态。式(4.23)为 Logistic 方程的一种演化形式[23]。

$$Cx_i^{(t+1)} = 4Cx_i^t(1-Cx_i^t), i = 0,1,2... \tag{4.21}$$

$$Cx_i = (x_i - a)/(b-a) \tag{4.22}$$

$$x_i^{'} = a + Cx_i(b-a) \tag{4.23}$$

其中，Cx_i^t 为混沌变量 Cx_i 在第 t 次迭代后的值;当 $Cx_i \in [0,1]$ 且 $Cx_i \in \{0.25,0.5,0.75\}$ 时，将产生混沌现象[24]。解空间变量 $x_i \in [a,b]$ 可通过式(6)和式(7)与 Cx_i 进行往返映射；a 和 b 分别为解空间的最大和最小值。

本文借鉴混沌现象随机性和遍历性的特点，提出了基于混沌优化摆脱局部极值的方法。该方法通过群体极值位置连续未更新的代数 SG 与局部极值判定阈值 SG_{max} 进行比较来判定算法是否陷入局部极值。若 $SG > SG_{max}$，则认为算法已经或即将陷入局部极值；反之，则认为算法没有陷入局部极值。当算法被判定为陷入局部极值时，首先利用式(4.22)把群体极值位置 G 映射到混沌变量定义域[0,1]内，然后利用式(4.21)进行迭代运算，得到 M 个混沌位置 $CG_1, CG_2, CG_3, ..., CG_m$，最后利用式(4.23)进行逆映射，获得 M 个新群体极值位置（$G^{1'}, G^{2'}, G^{3'}, ..., G^{m'}$）。由于粒子通过追逐个体和群体极值位置来完成自我更新，当算法陷入局部极值时，群体极值位置一定在局部极值位置上，此时，采用混沌映射得到具有较强随机性和遍历性的新群体极值位置，并结合式(4.17)就可以改变粒子的寻优轨迹，使得粒子 i 通过追逐新群体极值位置 $G^{i'}$ 进行自我更新时，可在局部极值位置邻域外的其他区域进行寻优搜索，搜索新的邻域和路径。因此可以较大概率地发现更优解，进而增加了算法摆脱局部极值的可能。

4.5.3 改进的独立自适应 PSO 算法

在进化过程中，粒子之间的进化能力，种群的整体进化能力以及所解决的问题是不同的，因此本文充分考虑了参数设置中的这些差异，从而使参数可以在不同情况下进行自适应调整。

定义 1 粒子演化能力：

在进化过程中，粒子进化能力被定义为粒子与其他粒子相比找到更好解决方案的能力，计算如下：

$$E_i^t = (p_{lworst}^t - f_i^t)/(p_{lworst}^t - p_{lbest}^t), \tag{4.24}$$

其中，f_i^t 代表第 t 代，第 i 个粒子的适应度值；p_{lworst}^t 表示在第 t 代值时所有粒子的最差适应度；因此，E_i^t 代表第 t 代粒子 i 的进化能力。

定义 2 种群演化能力：

在进化过程中，总体进化能力定义为总体中所有粒子找到比当前最优解更好的解决方案的能力。计算如下：

$$E_g^t = p_{gbest}^t - p_{gbest}^{t-1}, \tag{4.25}$$

其中 E_g^t 代表第 t 代种群的进化能力。

定义 3 进化率：

在进化过程中，粒子的进化速率定义为粒子在群体中进化能力的大小。粒子在种群中的进化能力的进化速率公式如下：

$$M_i^{t+1} = 1/\sqrt{(E_g^t)^2 + (E_i^t)^2 + 1}, \tag{4.26}$$

其中 M_i^{t+1} 代表粒子 i 在第(t+1)代的进化速率。

粒子在种群中的进化能力的演化速率计算如下：如果粒子演化能力和种群进化能力都强，则粒子演化速率小，则粒子将继承上一代粒子的进化能力，在下一代中更多。

- **自适应惯性权重**

在 PSO 算法中，惯性权重的设置决定了粒子的探索和搜索能力，这对算法的性能起着至关重要的作用。通过调整，可以调整粒子全局搜索和局部搜索功能的大小，以实现两者之间的平衡。如果 w 大，则粒子具有较强的全局搜索能力，否则，粒子具有较强的局部搜索能力。惯性权重的设置通常采用随着迭代次数的增加在一定间隔内线性减小的策略，其计算公式如下：

$$w = w_{max} - (w_{max} - w_{\min}) \times M_i^{t+1}, \tag{4.27}$$

其中 w_{max} 和 $w_{\min}$ 是惯性重量的上限和下限。

- **自适应性学习因子**

学习因子是两个参数，分别反映了粒子的历史最佳学习能力和群体的历史最佳学习能力。在本文中，将学习因子c_1设计为递减函数，并设计c_2为增加函数，以根据粒子进化能力的变化率来调整每一代中每个粒子的学习因子。与其他经典学习因子相比，这种设置方法不仅可以保证粒子在迭代初期的学习能力，而且可以增强全局搜索能力。以确保在迭代的后期对粒子进行社会学习，这有利于局部精确搜索，也可以通过独立粒子进行。使用不同的进化速率来调整学习因子，从而使粒子根据自己的条件调整学习模式。调整公式如下：

$$c_{1i}^{t+1} = c_{1\max} - \frac{c_{1\min} \bullet \sin(M_i^{t+1} \frac{\pi}{2}) \bullet t}{T}, \tag{4.28}$$

$$c_{2i}^{t+1} = c_{2\max} + \frac{c_{2\min} \bullet \sin(M_i^{t+1} \frac{\pi}{2}) \bullet t}{T}, \tag{4.29}$$

其中，$c_{1\max}$代表学习因子的最大值；$c_{2\max}$代表学习因子的最小值。粒子在迭代开始时应该具有很强的自学习能力。此时，c_1该值较大而c_1较小。如果粒子 i 的演化速率较小，则该粒子的c_1大小会比其他粒子大一点，c_2也要小一些，这更有利于粒子的全局搜索。随着迭代次数的增加，粒子应具有强大的社交学习能力。此时，c_1的值较小，而c_2的值较大。如果粒子 i 的演化速率较大，则该粒子比其他粒子具有较小的c_1和较大的c_2，这更有利于粒子的局部搜索。在本文中，根据粒子进化能力的变化率来自适应地调整每个粒子的学习因子。

4.5.4 基于 FastText 和 CPSO-XGBoost 的群众留言分类算法优化

- **模型构建**

根据 4.5.3 的实验结果比较，结合群众留言数据的相关特点，提出了一种基于 FastText 方法构建 150 维的词向量，以 XGBoost 算法作为群众留言分类器，并通过 CPSO 算法进行参数优化。基于 FastText 和 CPSO-XGBoost 的群众留言分类模型如图所示：

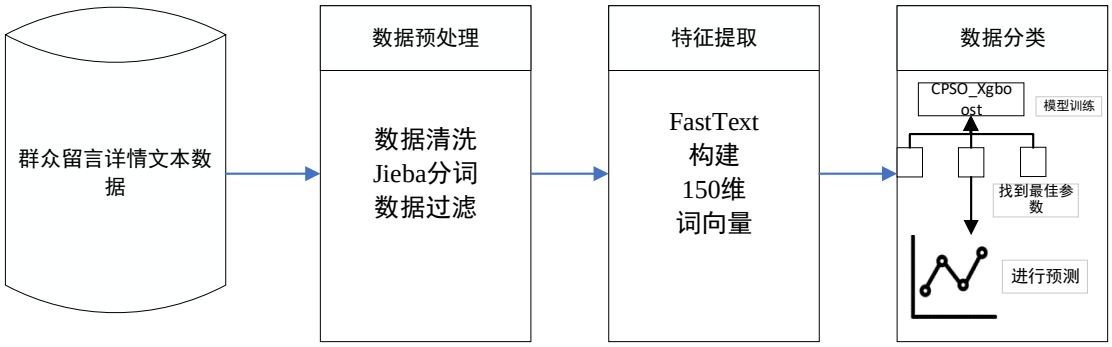


图 25 基于 FastText 和 CPSO-XGBoost 的群众留言分类模型

Figure 25 Classification model of Public Message based on FastText and CPSO-XGBoost

- **算法流程**

XGBoost 模型有 3 项主要参数，分别为学习率 learning_rate，树的最大深度 max_depth，最小叶子权重 min_child_weight，不同参数有不同功能。这些参数设定是否合理，对于模型的好坏有重要影响。调参通常取决于经验判断和遍历实验，传统方法效果不佳且缺乏理论依据。因此，本文基于改进的混沌粒子群算法进行参数优化，以(1-F1)值作为适应度函数，保留每次迭代的群体最优解和个体最优解，通过两者信息的交互，从而朝着全局最优进化。由于 PSO 陷入局部最优后无法跳出，CPSO 将粒子重新混沌映射后成功跳出局部最优，相比 PSO 表现出更好的全局寻优能力。基于 FastText 和 CPSO-XGBoost 的群众留言分类模型流程如下：

输入：算法迭代次数 T，种群规模 M，训练集数据 $Data_{train}$，测试集合数据 $Data_{test}$。

输出：最优位置 G，最优适应度函数值 V。

Step1. 对训练集数据 $Data_{train}$，测试集合数据 $Data_{test}$ 进行数据预处理，并通过 FastText 方法构建词向量。

Step2. 随机初始化种群中粒子的速度和位置，初始化迭代次数、计数器和局部极值判定阈值为 $t=0$、$SG=0$ 和 $SG_{max}=7$ [25]。

Step3. 定义 XGBoost 模型的相关参数,将粒子的初始位置传递给 XGBoost 模型的相关参数，作为 XGBoost 模型的初始参数，然后训练模型，以测试集的(1-F1)值作为粒子的初始适应度值 f，寻找种群的初始 p_{lbest} 和初始 p_{gbest}。

Step5. 对种群中所有粒子执行以下操作：①根据式(4.19)更新粒子的权重 ω_{t_i} 。②根据式(4.17)和式(4.18)更新粒子速度和位置。③计算粒子适应度值 f，并更新粒子的 p_{lbest} 和 p_{gbest}，如果寻优迭代次数过半，p_{gbest} 未发生更新，$SG=SG+1$，否则 $SG=0$ 。

Step6. 如果 $SG \geq SG_{max}$，利用式(4.21)~(4.23)对 G 进行混沌优化生成（$G^{1'},G^{2'},G^{3'},\ldots,G^{m'}$），$SG=0$ 。

Step7. $t=t+1$，如果 $t<T$，转(4)，否则，执行(7)。

Step8. 输出 G，V，算法结束。

- **实验与分析**

根据改进的混沌粒子群算法的特点，结合 XGBoost 的参数范围以及群众留言数据的性质进行相应设定。设置初始种群数量为 40，每个个体包含 3 个参数，参数在待选范围内随机生成,迭代 100 次，优化过程如图所示：

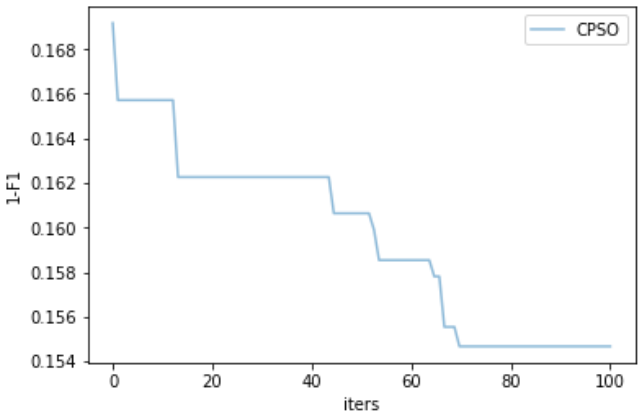


图 26 CPSO 算法寻优过程

Figure 26 CPSO Algorithm Optimization Process

根据图，发现在第 52 次和第 61 次迭代时通过混沌映射成功跳出局部极值，找到了更优解。

为了验证基于 FastText 和 CPSO-XGBoost 的群众留言分类模型的有效性，在群众留言分类中将此模型与各常用模型进行对比分析。采用 3 项指标评估算法优劣。ACC 表示分类正确率，macro F1 表示 F1 值宏平均，micro F1 表示 F1 值微平均，它们是分类任务中最常用的性能指标，值越高代表分类越准确。各算法对比结果如表 6 所示。

表 6 算法对比

Table 6 Algorithm Contrast

算法	macro F1	micro F1	ACC

FastText(dim=150)-CPSO-XGBoost	0.845329	0.860803	0.860803474
FastText(dim=100)-XGBoost	0.828719	0.8519	0.851900109
FastText(dim=150)-XGBoost	0.834403	0.85342	0.853420195
Docvec(dim=300)-XGBoost4	0.802263	0.826927	0.826927253
Docvec(dim=100)-XGBoost5	0.786163	0.815635	0.815635179
Word2vec(dim=100)-XGBoost	0.80157	0.823453	0.823452769
Word2vec(dim=300)-XGBoost	0.794972	0.817807	0.817806732
TF-IDF-XGBoost	0.813274	0.819978	0.819978284

从表 6 中发现，在针对群众留言分类问题上 FastText 方法构建 150 维词向量与 Docvec、Word2vec 和 TF-IDF 对比，在三个指标方面都表现更优。经过 CPSO 算法优化的基于 FastText 和 CPSO-XGBoost 的群众留言分类模型不仅提高了分类正确率，同时在 F1 值方面也有所增强，分别提升了 **0.86%，1.29%**。

4.5.5 基于 TF-IDF 和 IAPSO-SVM 的群众留言分类算法优化

- **模型构建**

根据 4.5.3 的实验结果比较，结合群众留言数据的相关特点，提出了一种基于 TF-IDF 方法选取前 3000 的关键字构建 VSM，以 SVM 算法作为群众留言分类器，并通过 CPSO 算法进行参数优化。基于 TF-IDF 和 IAPSO-SVM 的群众留言分类模型如图所示：

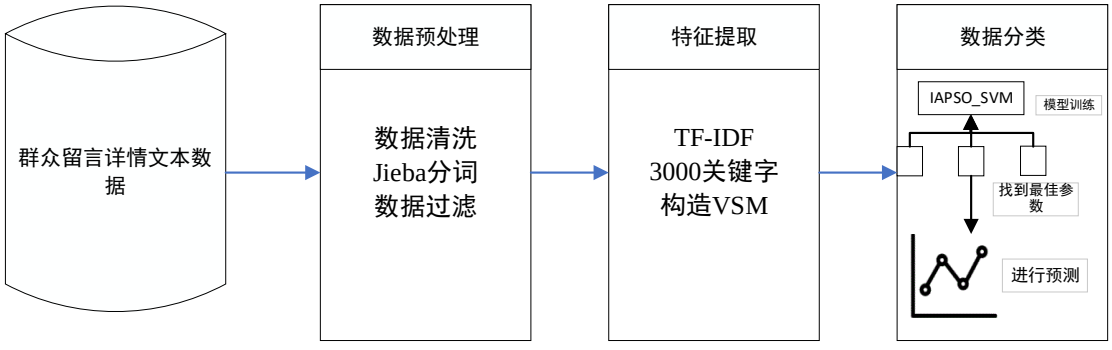


图 27 基于 TF-IDF 和 IAPSO-SVM 的群众留言分类模型

Figure 27 Classification model of Public Message based on TF-IDF and IAPSO-SVM

- **算法流程**

本文采用线性核的 SVM 算法，主要优化 SVM 算法的误差项的惩罚参数 C。参数 C 设定是否合理，对于模型的好坏有重要影响。调参通常取决于经验判断和遍历实验，传统方法效果

不佳且缺乏理论依据。因此，本文基于改进的混沌粒子群算法进行参数优化，以(1-F1)值作为适应度函数，保留每次迭代的群体最优解和个体最优解，通过两者信息的交互，从而朝着全局最优进化。IAPSO-SVM 算法的主要步骤如下：

Step.1 随机初始化 N 个粒子，初始化参数 w_{max}， $w_{\min}$， $c_{1\max}$， $c_{2\max}$， $c_{1\min}$， $c_{2\min}$，最大迭代次数 T， 问题维数 D 等。

Step.2 计算粒子的适应度值，找到个体最优 p_{lbest} 和全局最优 p_{gbest}；

Step.3 根据公式(4.17)和(4.18)更新粒子速度和位置。

Step.4 计算粒子的适应度值，更新个体最优和全局最优；

Step.5 根据公式(4.24)-(4.29)计算生成时粒子的惯性权重和学习因子

Step.6 确定算法是否达到终止条件。如果满足，则算法停止并输出最佳值。如果不满意，请跳至步骤 3 以继续执行。

- **实验与分析**

根据 IAPSO 算法的特点，结合 SVM 的参数范围以及群众留言数据的性质进行相应设定。设置初始种群数量为 20,每个个体包含 1 个参数,参数在待选范围内随机生成,迭代 100 次，优化过程如图所示：

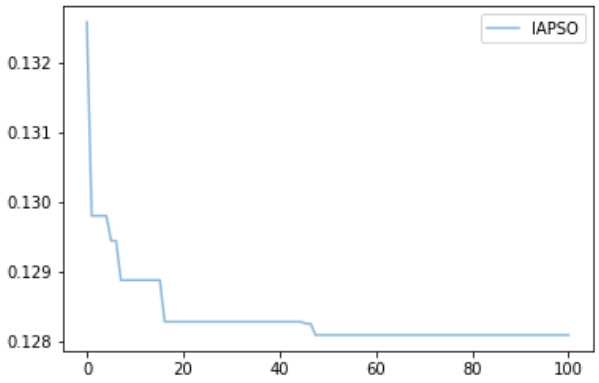


图 28 IAPSO 算法寻优过程

Figure 28 IAPSO Algorithm Optimization Process

为了验证基于 TF-IDF 和 SVM 的群众留言分类模型的有效性，在群众留言分类中将此模型与各常用模型进行对比分析。采用 3 项指标评估算法优劣。ACC 表示分类正确率，macro F1 表示 F1 值宏平均，micro F1 表示 F1 值微平均，它们是分类任务中最常用的性能指标，值越高代表分类越准确。各算法对比结果如表 7 所示。

表 7 算法对比

Table 7 Algorithm Contrast

算法	macro F1	micro F1	ACC

TF-IDF-IAPSO-SVM	0.8719148546927196	0. 878610206298	0.878610206298
TF-IDF-SVM	0.870497463	0.877741585	0.877741585
LDA-SVM	0.477445151	0. 571769815	0.571769815
Word2vec(dim=300)-SVM	0.854183673	0. 868403909	0.868403909
Word2vec(dim=100)-SVM	0.84104519	0. 859066232	0.859066232
FastText(dim=100)-SVM	0.839223607	0. 856677524	0.856677524
FastText(dim=150)-SVM	0.846790804	0. 863409338	0.863409338
Docvec(dim=100)-SVM	0.82257431	0.842996743	0.842996743

从表 7 中发现，在针对群众留言分类问题上 TF-IDF 方法选取前 3000 的关键字构建 VSM 与 Docvec、Word2vec、FastText 和 LDA 对比，在三个指标方面都表现更优。经过 IAPSO 算法优化的基于 TF-IDF 和 IAPSO-SVM 的群众留言分类模型不仅提高了分类正确率，同时在 F1 值方面也有所增强，在 macro F1 达到 **0.871915**，ACC 也达到了 **0.878610207。**

4.6 带类别权重的卷积神经网络群众留言分类方法

- **模型构建**

根据 4.4 的实验结果比较，结合群众留言数据的主题不均衡现象，提出了一种基于一种带类别权重的 loss 函数来优化卷积神经网络。带类别权重的卷积神经网络的群众留言分类模型如图 29 所示：

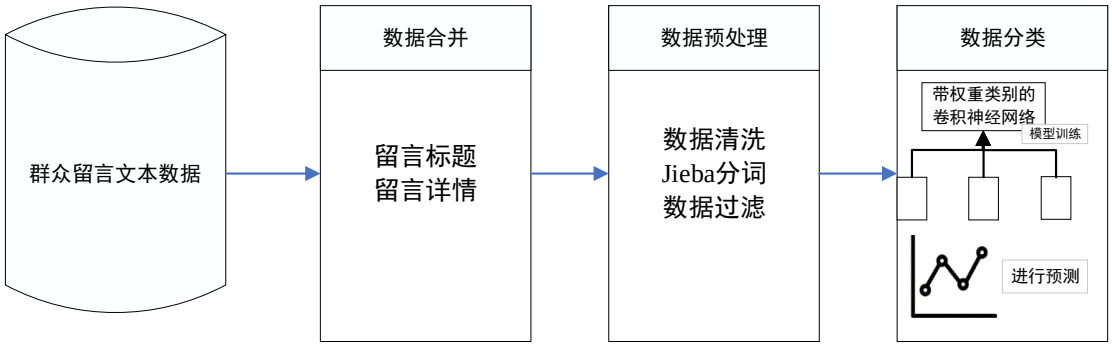


图 29 基于 FastText 和 CPSO-XGBoost 的群众留言分类模型

Figure 29 Classification model of Public Message based on FastText and CPSO-XGBoost

- **带类别权重的卷积神经网络**

神经网络的训练过程中根据算法对其进行更新，先正向计算得到网络输出误差，然后反向更新网络权重，使网络的输出误差最小，本文采用批量梯度下降的方法更新网络参数。神经网络输出结果为$\mathrm{x} = [x_1, x_2, \ldots, x_N]$，$x \in \mathbb{R}^{\mathrm{N}\times\mathrm{M}}$，其中$N$是 batch size 包含的样本个

数，M是输入数据集类别总数。

误差函数为交叉熵损失函数E，刻画的是实际输出(概率)与期望输出(概率)分布的距离，是编译一个神经网络模型的重要要素。定义为

$$E(x，\ label) = -w_{label}\log\frac{e^{x_{label}}}{\sum_{j=1}^{N}x_j}$$

其中，w_{label}是不同类别的权重，默认为 1。

我们对标签权重重新赋值为$w = [w_1，\ w_2，\ \ldots，\ w_N]$，其中$w_j$为第$j$类标签对应的标签权重，具体计算方式为:

$$w_j = 1 + \frac{T}{d_j}$$

其中，T是超参数，我们取为 0.1，d_j为第j类标签在训练集中的文本总数，可以看出标签权重与标签所有的文本总数成反比.在不平衡数据集上,多数类与少数类样本数相差较大，如果类标签权重与标签所有的文本总数成反比，这样可以强化少数类对模型参数的影响，让神经网络对少数类更加敏感，从而获得更好的分类效果。

- **实验与分析**

为了验证带类别权重的卷积神经网络的有效性,在群众留言分类中将此模型与各常用模型进行对比分析。采用 3 项指标评估算法优劣。ACC 表示分类正确率，macro F1 表示 F1 值宏平均，micro F1 表示 F1 值微平均，它们是分类任务中最常用的性能指标，值越高代表分类越准确。各算法对比结果如表 8 所示。

表 8 算法对比

Table 8 Algorithm Contrast

算法	macro F1	micro F1	ACC
Weight_TextCNN	**0.906972567**	**0.910966341**	**0.910966341**
TextCNN	0.900545333	0.903365907	0.903365907
TF-IDF-IAPSO-SVM	0.8719148546927196	0. 878610206298	0.878610206298
FastText(dim=150)-CPSO-XGBoost	0.845329	0.860803	0.860803474

从表 8 中发现，在针对群众留言分类问题上带类别权重的卷积神经网络方法与 TF-IDF-IAPSO-SVM 和 FastText(dim=150)-CPSO-XGBoost 对比，在三个指标方面都表现更优。经过带类别权重优化的卷积神经方法较基于 4.5.4，4.5.5 节优化后的群众留言分类模型分类正确率分别提高了 **3.55%,5.51%**,同时在 F1 值方面也有所增强分别提高了 **3.87%,6.80%。**

4.7 小节

本章首先对各种文本表示方法与机器学习算法进行了组合实验，发现基于 FastText-XGBoost 和 TF-IDF-SVM 算法在文本分类中表现出色,F1 分别达到 0.834402595,0.870497463。接着对深度学习模型在群众留言数据进行了实验，发现 TextCNN，DPRCNN，Fasttext 方法具有良好的表现，在 F1 分别达到 0.900545333，0.895310849，0.885583369。随后进一步对基于 FastText-XGBoost 和 TF-IDF-SVM 算法在群众留言分类展开研究，分别提出了两种进化算法——CPSO 算法和 IAPSO 算法对模型进行优化并应用。CPSO-XGBoost 算法较原模型在 macro F1 和 ACC 分别提升了 1.29%，0.86%，IAPSO-SVM 的 macro F1 达到 0.871915，ACC 也达到了 0.878610207。对于深度学习模型，我们提出了一种带类别权重的卷积神经网络模型，相较优化后的机器学习算法在 F1 提升了 3.55%，5.51%，在 ACC 上提升了 3.87%，6.80%，带类别权重的 TextCNN 模型相较其他深度学习模型，不仅提高了分类正确率，同时在 F1 值方面也有所增强，macro F1 达到 0.906972567，ACC 也达到了 0.910966341。

第五章 问题二

话题发现和跟踪是指新闻专线和广播新闻等来源的新闻数据流中自动地发现话题并把话题相关的内容组织到一起的技术。通过增量的文档聚类的方法，信息流被聚集到有限的话题类簇中，类内高度相似，不同的类间相似度较低，以此进行海量数据的融合。

热点发现是在各话题舆情中发现，关注度较为大，影响也较为突出的舆情事件，旨在从半结构化海量数据中获取相应的主题并进行整合，以新的热点事件分析并了解热点话题事件，可及时向政务部门反馈，辅助政务处理。

热点发现需要将留言划分为不同的话题圈，并根据各项指标计算话题热度值，其中，热度值高的话题被认为是热点问题。本章就这一过程进行阐述。

5.1 基于 Singlepass 算法的话题分类

5.1.1 文本数据处理

关键词提取

提取热点问题,首先需要对留言文本进行分词处理,分词过后的文本能够过滤掉很多“停用词”，保留下“关键词”，使得一个句子能够被少数几个“关键词”代表，从而数据变得简略，更易于下游任务执行。本文使用 Jieba 做去停用词，提取关键词处理，采用 TF-IDF 算法调整语料中不同词的词频，将那些在所有文档中都出现的高频词的权重降低。

词向量转换

在进行聚类之前需要将各关键词文本分别在它们内部转化为特征向量，才能进行聚类。本文按照类别将文本输入程序转化为 one-hot 编码的特征向量[26]。

One-hot 词向量中只有一个维度值为 1，剩下的维度值为 0，维度值为 1 的位置说明了该词在词表中的位置。例如，“噪音”的 one-hot 词向量表示为[0 0 0 0 1 0 0 0 0 0 0 …]，说明噪音为词表的第四个词。由于 one-hot 词向量中大部分的维度值为 0，所以利用 one-hot 表示的词向量具有稀疏性,而对于一条留言,其向量为包含关键词的向量之和。

聚类

得到分词过后的文本,需要对文本进行聚类,聚类后的文本在各自类中有很高的相似性,至此，各类之中就已经出现了热点问题的雏形。本文采用 Single-Pass 聚类算法的思想并结合相似度系数计算文本相似度，对文本进行聚类。

计算词特征向量的余弦相似度。余弦相似度，又称为余弦相似性，是通过计算两个向量的夹角余弦值来评估他们的相似度，我们据此计算词向量间相似度，依据相似程度判断两个向量是否聚为一类。

假设 y(A)和 y(B)分别是两条留言的词向量，则它们的余弦相似度计算如下式所示。

$$R = \frac{y(A)^T y(B)}{|y(A)||y(B)|}$$

具体判别流程为：

Step1. 输入第一句话 s1 设定一个初始类别 A 并将 s1 放入 A 类。

Step2. 输入第二句话 s2 根据“Single-Pass 聚类规则”判定 s2 与 s1 是否相似，若相似则将 s2 归入 A 类，若不相似则生成新的类 B，将 s2 归入 B 类。

Step3. 输入第三句话用相同的聚类规则让 s3 先 s1 对比，若 s3 与 s1 相似则 s3 归入 s1 所在类，否则将 s3 与 s2 对比，若 s3 与 s2 相似则 s3 归入 s2 所在类，若 s3 与 s1、s2 都不相似则生成新的类 C，将 s3 归入 C 类。

Step4. 后续向量也按照此方法进行归类，最终得到聚类结果提取出的众多文本集合即为所求的热点问题。

上述判别流程中的“Single-Pass 聚类规则”为：计算两两特征向量的余弦相似度，根据特征向量长度设定阈值，若 A，B 两个向量之间相似度大于等于阈值则两向量聚为一类。

据此，对附件三中留言转换后的词向量做聚类处理，划分为不同的话题圈。

5.1.2 话题划分结果

对附件中 4326 条留言进行基于 Singlepass 的话题分类处理，共分出 383 个话题，详情话题划分结果见附件，现对话题划分结果做统计分析。

其中：

表 9 基于 Singlepass 话题划分结果

Table 9 Topic Division Results based on Singlepass

话题下留言数	话题数
>100	10
(100，50]	5
（50，10]	23
(10，5]	31

（4，2]	64

再对各话题主题关键词数作统计，结果如下：

表 10 基于 Singlepass 话题划分关键词数统计

Table 10 Statistics of Keyword Number based on Singlepass

话题下主题关键词数	话题数
34	34
6	3
5	1
4	4
3	30
2	38
1	22
0	1

可见，基于 Singlepass 的话题划分粒度不够细致，有些话题的主题关键词仅为一个名词且多以地名作为区分。导致热点计算时，热度值会偏向留言数目多的话题，而留言间区分度不够使这并不是理想的结果。

5.2 基于 LDA 模型的话题分类

Blei 等人提出了 LDA 模型（Latent Dirichlet Allocatyion），并把 LDA 模型描述成“离散数据文本的一个生成概率模型”[27]。LDA 是一个三层的贝叶斯模型，该模型的基本思想是：集合中的每篇文档代表了潜在主题所构成的一个概率分布，而每个主题又代表了很多词汇所构成的一个概率分布。该模型可以对海量文档集进行建模，并可以将文档表示为由特定数目的潜在主题信息组成。图 29 给出了模型的拓扑结构。

5.2.1 文本数据处理

在 2.1.1 中已对留言文本做提取关键词，词向量转换处理，基于 LDA 模型需要做相同的数据处理，故在本小节不再赘述。

5.2.2 主题生成

令K为主题的数目，V为词表的大小，α和η为正实数，$Dir_K(\cdot)$为K维狄利克雷分布，则有 LDA 的生成过程如下：

1. 对每个主题k

(a) 抽取其在词表上的分布$\beta_k \sim Dir_V(\eta)$

2. 对条留言d

a) 抽取留言d上的主题分布$\theta_d \sim Dir_K(\alpha)$

b) 对于留言d中的第n个词$w_{d,n}$

i. 抽取其对应的主题$z_{d,n} \sim Mult(\theta_d)$，其中，$z_{d,n} \in \{1,\dots,K\}$

ii. 抽取词$w_{d,n} \sim Mult(\beta_{d,n})$，其中，$w_{d,n} \in \{1,\dots,V\}$

对于一篇包含个N个词的留言，主题的混合比例θ，留言中每个词分配的主题的集合，以及词语集w在α,β条件下的联合概率分布如式(5.1)所示。

$$p(\theta,z,w|\alpha,\beta) = p(\theta|\alpha)\prod_{n=1}^{N} p(z_n|\theta)p(w_n|z_n,\beta) \tag{5.1}$$

通过对θ积分以及对所有z进行求和，可以得到该留言的边缘分布如式(5.2)所示。

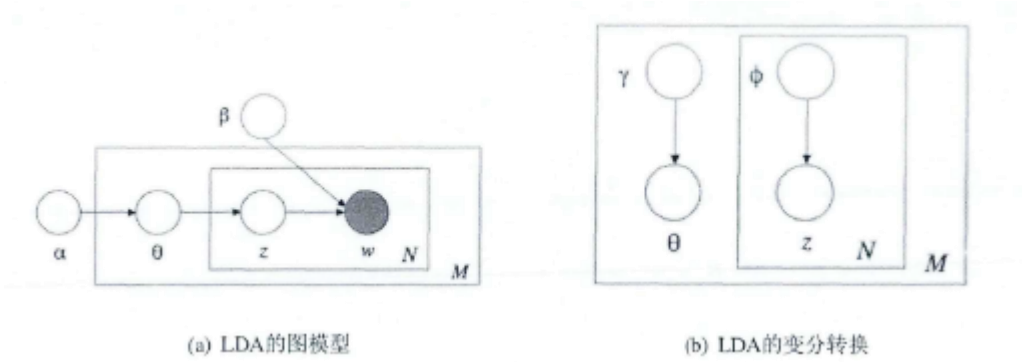
$$p(w|\alpha,\beta) = \int p(\theta|\alpha)\left(\prod_{n=1}^{N}\sum_{z_n} p(z_n|\theta_n)p(w_n|z_n,\beta)\right)d\theta \tag{5.2}$$

最后，将留言集中的每篇留言的边缘分布求积，便可得到整个留言集的概率分布，如式(5.3)所示。

$$P(D|\alpha,\beta) = \prod_{d=1}^{M}\int p(\theta_d|\alpha)\left(\prod_{n=1}^{N_d}\sum_{z_{d,n}} p(z_{d,n}|\theta_n)p(w_{d,n}|z_n,\beta)\right)d\theta \tag{5.3}$$

5.2.3 参数估计

由于狄里克雷分布θ_d和β_k参数值是无法直接获得的，我们使用变分贝叶斯进行近似的参数估计[28]。



(a) LDA的图模型　　(b) LDA的变分转换

图 30 LDA 变分推导的转换

Figure 30 Transformation of LDA Variational Derivation

从图 5.6(a)可以发现，引发θ和β耦合的关键是θ、z和w之间的连线，去掉这些连线和w节点，通过由此得到具有无约束的变分参数的简化的图模型（图 29(b)），我们可以获得在隐变量上的分布，如式(5.4)所示。

$$q(\theta,z|\gamma,\phi)=q(\theta|\gamma)\prod_{n=1}^{N}q(z_n|\phi_n) \tag{5.4}$$

其中，狄利克雷参数γ和多项式参数$(\phi_1,\ldots,\phi_N)$是无约束的变分参数。

由此，我们就获得了一个可认为与原分布等价的新的分布，也即变分分布。接下来通过最小化原分布和变分分布之间的 KL 测度来求解γ和ϕ值的优化问题，如式(5.5)所示。

$$(\gamma^*,\phi^*)=\arg\min_{\gamma,\phi}D(q(\theta,z|\gamma,\phi)||p(\theta,z|z,\alpha,\beta)) \tag{5.5}$$

则有式(5.6)

$$\begin{aligned}L(\gamma,\phi;\alpha,\beta)=&E_q[\log p(\theta|\alpha)]+E_q[\log p(z|\theta)]\\&+E_q[\log p(w|z,\beta)]-E_q[\log q(\theta)]-E_q[\log q(z)]\end{aligned} \tag{5.6}$$

为了获得使$L(D)$最小的参数值，可以通过期望最大化（EM）算法来将各式带入后通过计算对应的偏导并且使之等于θ，可以得到式(5.7)。

$$\begin{aligned}\phi_{ni}&\propto\beta_{iw_n}\exp\{E_q[\log(\theta_i)|\gamma]\}\\\gamma_i&=\alpha_i+\sum_{n=1}^{N}\phi_{ni}\end{aligned} \tag{5.7}$$

多项式的期望的可以通过式(5.8)进行更新。

$$E_q[\log(\theta_i)|\gamma]=\Psi(\gamma_i)-\Psi\left(\sum_{j=1}^{k}\gamma_j\right) \tag{5.8}$$

其中，Ψ作为的$\log\Gamma$一阶导数可通过泰勒公式求得。

EM 算法的 E 过程如算法 1 所示，M 过程的参数估计涉及α和β两个参数，通过$\frac{\partial L}{\partial \beta_{ij}}=0$，有式(5.9)。

$$\beta_{ij} \propto \sum_{d=1}^{M}\sum_{n=1}^{N_d} \phi_{d_{ni}} w_{dn}^{j} \tag{5.9}$$

对于α可以得到其偏导，如式(5.10)所示。

$$\frac{\partial L}{\partial \alpha_i} = M\left(\Psi\left(\sum_{j=1}^{k}\alpha_j\right)-\Psi(\alpha_i)\right)+\sum_{d=1}^{M}\left(\Psi(\gamma_{d_i})-\Psi\left(\sum_{j=1}^{k}\gamma_{d_j}\right)\right) \tag{5.10}$$

用牛顿-辛普森算法来迭代求解α，其中要用到的海瑟矩阵可以用式(5.11)得到。

$$\frac{\partial L}{\partial \alpha_i \alpha_j} = \delta(i,j)M\Psi'(\alpha_i)-\Psi'\left(\sum_{j=1}^{k}\alpha_j\right) \tag{5.11}$$

由此，变分算法通过在 E 和 M 两个步骤间进行不断的迭代，直到对数似然的下限收敛为止。

算法 EM 变分的 E 步骤

Initialize $\phi_{ni}^{0} := 1/k$ for all I and n

Repeat

For n=1→n do

For l=1→n do

$\phi_{ni}^{t+1} := \beta_{iw_n}\exp\{\Psi(\gamma_i^t)\}$

Normalize ϕ_n^{t+1} to sum to 1

$\gamma_i^{t+1} := \alpha + \sum_{n=1}^{N}\phi_n^{t+1}$

Until convergence

5.2.4 确定最优主题数

在 LDA 主题模型中采用困惑度指标评定法，将困惑度最小的主题数量作为最优主题数量。随着主题数的增加，模型的困惑度逐渐降低；但主题数越多，模型中每个主题的内容也越难确定，并且主题与主题之间的差别也会越来越小。在实际应用中，一般选取困惑度在逐渐降低并趋于平缓点的“拐点”处的主题数。[29]

困惑度计算公式为式(5.12)

$$perplexity = \exp\left\{-\frac{\left(\sum_{m=1}^{M}\sum_{n=1}^{N_m}\log\left(\sum_{k=1}^{K}p(w_n|z_k)\,p(z_k|d_m)\right)\right)}{\sum_{m=1}^{M}N_M}\right\} \quad (5.12)$$

其中，M为数据集文本数，N_m为第M篇文本词项总数。

5.2.5 话题划分结果

困惑度计算结果如下：

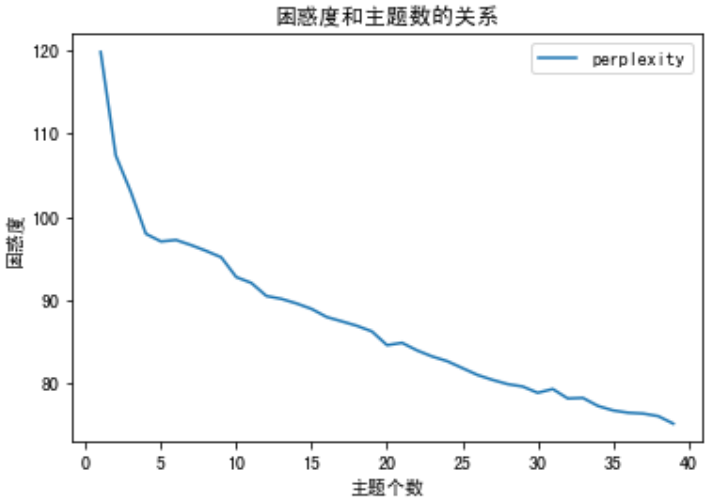


图 31 困惑度与主题数关系图

Figure 31 Relation between Perplexity and the Number of Topics

根据困惑度，确定出最优话题数目，定为 20。根据确定话题数，即可确定话题划分结果，结果如下所示，具体结果见附件：

表 11 基于 LDA 话题划分结果

Table 11 Topic Division Results based on LDA

主题编号	主题数目
1	84
2	33
3	79
4	54
5	31
6	44
7	29

8	42
9	29
10	55
11	45
12	29
13	63
14	33
15	51
16	43
17	47
18	36
19	36
20	40

根据划分结果，可见 LDA 划分话题粒度更细，话题区分度均匀，划分效果更理想，各话题间相似度更高。故热度计算时选择 LDA 模型得出的话题区分结果。

5.3 热度计算

话题划分结束需要考虑话题热度，综合考虑话题下留言数目，留言数目增长率，点赞总数，反对总数四个信息，定义热度计算方法。

5.3.1 热度计算指标

1) 留言数目

留言是人民反映问题，提出需求的有效途径，多条留言代表着多数人的诉求，故留言数目是反映问题热度的重要指标。同一话题下留言数目，是多人反映的同一问题的有效衡量值。

2) 留言增长率

一段时间集中爆发的问题也是需要密切关注的热点问题。为衡量问题的爆发程度，引入留言数目增长率这一特征。

留言时间从 2017 年 6 月至 2020 年 01 月，时间跨度较大，故以月为单位，划分时间窗，按月计算留言数目增长率，按条计算话题下留言总增长率，求和记为该话题增长率，计算公

式为：

$$F_{ij} = \frac{f_{ij}}{1 + f_{i,j-1}} \tag{5.13}$$

其中，F_{ij}表示话题 T_i 下留言在时间窗 j 内的增长率，f_{ij}为话题 T_i在时间窗 j 内的留言数.

话题T_i增长率：

$$S = \sum_j F_{ij} \tag{5.14}$$

3) 点赞数

网友点赞意味着此条留言得到了他人的认同，是对留言内容真实性的肯定，或是对留言反映问题的共鸣。无论是哪种情况，都说明了群众对此条留言群众的认可，点赞数是话题热点的重要指标。

4) 反对数

反对有可能是此留言描述内容有误，网友对留言真实性的质疑，意味着此条留言有争议，则更应该成为需要关注的热点问题。

5.3.2 热度计算方法：

综合考虑四项指标，计算话题热度。记话题T_i下留言数目为X_1，留言增长率X_2，点赞数X_3，反对数X_4。对X_1，X_2，X_3，X_4做归一化处理后得到X_1'，X_2'，X_3'，X_4'，热度计算公式为：

$$H = X_1' + X_2' + X_3' + X_4' \tag{5.15}$$

归一化函数为

$$X_i' = \left(\frac{2}{1 + a_i^{-X_i}} - 1 \right) * w_i \tag{5.16}$$

$$X_i' = \left(\frac{2}{1 + a_i^{-X_i}} - 1 \right) * w_i \tag{5.17}$$

其中，标准化公式参数

$$a_i = \begin{cases} 1.1, \max(x_i) = 50 \\ 1.01, \max(x_i) = 500 \\ 1.005, \max(x_i) = 2000 \\ 1.001, \max(x_i) = 5000 \end{cases}$$

权重系数$w_1 = 40\%, w_2 = 45\%, w_3 = 10\%, w_4 = 5\%$

5.4 热点发现结果

根据热度计算公式，计算各话题热度值，各话题热度值如下所示：

表 12 话题热度值

Table 12 Topic Heat Value

话题编号	X_1'	X_2'	X_3'	X_4'	S
1	0.1883446	00.99689	0.715025	0.9801119	0.8279457
2	0.047619	0.853759	0.599642	0.839021	0.697625
3	0.278385	0.995434	0.597878	0.903047	0.771442
4	0.047619	0.96914	0.696737	0.966071	0.800176
5	0.047619	0.831475	0.499641	0.994461	0.659256
6	0.141999	0.934546	0.618131	0.989133	0.757991
7	0.047619	0.80615	0.414785	0.866621	0.598156
8	0.695118	0.924068	0.603695	0.794139	0.75546
9	0.233866	0.80615	0.427386	0.722756	0.598753
10	0.404398	0.971393	0.601383	0.998243	0.779224
11	0.047619	0.939247	0.55849	0.413463	0.670747
12	0.095023	0.80615	0.4062	0.975691	0.60757
13	0.278385	0.984442	0.593723	0.98984	0.773856
14	0.141999	0.853759	0.454259	0.925175	0.645537
15	0.233866	0.961281	0.510627	0.998243	0.725812
16	0.781188	0.929495	0.609552	0.914794	0.776635
17	0.188345	0.947688	0.557709	1	0.739461
18	0.047619	0.882132	0.465442	0.914794	0.656162
19	0.047619	0.882132	0.444752	0.958593	0.651232
20	0.095023	0.911989	0.578454	0.99112	0.728963

其中，热度中高的话题分别为：话题一，话题四，话题十，话题十六，话题十三。

表 13 前五话题

Table 13 top five topics

话题编号	热度排名	热度指数	时间范围	地点/人群	问题描述
1	1	0.8279457	2019/1/4 至 2020/1/26	A 市 A1 区中房瑞致的小区； A 市 A2 区的丽发新城小区,文源社区南苑花园小区； A 市 A3 区枫林路,洋湖街道莲香园小区,学丰路卓越千山；	小区工地施工,噪声扰民
4	2	0.800176	2019/1/1 至 2019/12/20	A 市 A1 区	小区非法饭店油烟噪音扰民
10	3	0.779224	2019/1/5 至 2019/12/23	A 市一中 A5 区；A6 区原高塘岭镇	交通拥堵,公交
16	4	0.776635	2019/1/5 至 2019/12/19	A 市	基础设施建设
13	5	0.773856	2019/1/13 至 2019/12/30	A 市万科魅力之城小区	墙面开裂

第六章 问题三

2020 年 1 月江苏省人民政府办公厅积极履行法定主管部门职责，研究制定了《江苏省政府信息公开申请办理答复规范》,从内容对答复文本作规定,然而并未有一个全国范行的，完整的，统一的答复评价标准。

为答复意见做质量评价对提升政务服务水平具有重要意义，有助于规范答复文本格式，促进答复内容更完整，更全面，更利于民众理解，除此，对答复文本作分析也有助挖掘如今政务答复普遍存在问题，促进政务服务优化。为此，给出一套政务答复意见评价体系，就本章就此展开阐述。

6.1 答复意见特征提取

对答复文本定义相似性，完整性，可解释性，及时性四个特征，分别从内容，格式，合理程度，时间这四个角度对文本进行评价。

6.1.1 答复意见相似性

为描述答复意见与留言的相关程度，即主题的契合程度，要求答复意见描述内容与留言所描述的必须是同样一个事件，引入答复意见相似度概念。相似性用于描述答复与留言的文本相似程度，刻画答复内容的准确度。例如，答复主题与留言主题相去甚远，认为这是相似度低的，不可靠的答复。

- **评价模型**

针对于留言与回复属于长文本特性，依据关键词计算余弦相似度，计算留言与回复相似程度。留言包括正文，标题，标题在一定程度上反映了留言主题，正文是留言的主要部分。故用留言标题与留言正文分别于答复计算文本相似度，结果做线性加权，得到最终的相似度数值。

留言有重述标题的情况存在，如“市民同志：你好！您反映的“请加快提高民营幼儿园教师的待遇”的来信已收悉。现答复如下：”，之后才是正式的答复，因此，计算相似度时将此类，不属于正式答复的内容去掉，否则影响相似度结果。

- **算法流程**

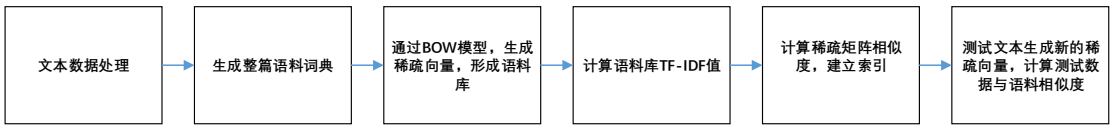


图 32 文本相似度计算流程

Figure 32 Text Similarity Calculation Process

文本预处理：

1) 将留言标题，留言正文，通过 jieba 分词工具分词，去停用词，词性过滤（仅保留对主题贡献度大的名词与动词）处理后，形成一个二维数组，得到文档的特征列表。

2) 生成整篇语料的词典；

3) 通过 BoW 模型将文本生成为稀疏向量。Bag-of-words model (BoW model，词袋模型)使用一组无序的词语来表达一段文字或一个文档；

主题向量变化，通过挖掘预料中蕴藏的语义结构特征，最终变换出一个简洁高效的文本向量。

4) 稀疏矩阵加权处理。TF-IDF（term frequency－inverse document frequency，词频-逆向文件频率）算法是一种统计方法，用以评估一个字词对于一个语料库中的其中一份文本的重要程度。 TF-IDF 认为一个词语出现的文本频数越小，它区别不同类别文本的能力就越大。因此引入了逆文本频度 IDF 的概念，以 TF 和 IDF 的乘积作为特征空间坐标系的取值测度，并用它完成对权值 TF 的调整，调整权值的目的在于突出重要词语，抑制次要词语。选用 TF-IDF 算法，计算得到语料库的 TF-IDF 值，用于描述词语的重要程度，给稀疏矩阵加权；

得到每篇文档对应的主题向量后，计算文档之间的相似度。

5) 计算稀疏矩阵相似度，建立索引

6) 对留言内容做同样的预处理操作，得到稀疏向量，将语料库训练结果中的向量表示与留言内容的向量表示做矩阵相似度计算，即求得文本相似度。

7) 计算留言内容与答复的文本相似度，留言标题与答复的文本相似度，求和，作为该答复相似度

6.1.2 答复意见完整性

为提高可读性，可理解性，便于发帖人阅读，答复意见不能是毫无章程的，混乱的，如同书信，网络电子邮件，答复意见应该满足一套格式标准，这个标准使得答复意见清晰可阅。为评价答复文本格式的规范性，建立标准答复格式，引入答复意见完整性概念，完整性用于

描述答复文本的规范程度。

- **评价模型**

如同书信，答复文本格式应该尽可能规范。一般书信可分为称呼，正文，结尾，署名，日期五个部分。

完整的答复应有：

- 起首语，加上“尊敬的”、“敬爱的”、“亲爱的”等形容词，以表示敬重或亲密之情；
- 问候语，祝颂语，如“你好”，“祝您生活愉快”；
- 署名。结尾处标注答复人姓名；
- 日期。结尾处应标注答复日期，且日期应位于署名后。

从政务答复特点的角度出发，完整的答复应有：

- 类似于摘要，答复中，需要表示已收到市民的留言，标明正式答复内容，例如 “您反映的:“XXXX”问题收悉,现答复如下:”

综合考虑书信，网络回帖，政务答复特征，建立规范答复格式如下：

亲爱的网友（称呼）：

您好！（问候语）

您所反映“XXX”的问题已收悉。现答复如下：(已收到)

XXXX（正文）

感谢您对我们工作的信任与支持！（祝颂语）

XXX 单位（署名）

XXXX 年 XX 月 XX 日（日期）

根据分析给出计算方法：

1) 起首语

内容上，标准起首语应为“形容词“+”称呼”，形容词以表示亲密之情，称呼指明收件人，如“亲爱的市民”，据此建立起首语标准语料库。

考虑到起首语文本长度短，内容简单的特点，故采用编辑距离（Edit Distance，ED）算法计算分数，编辑距离指的是在两个单词之间，由其中一个单词转换为另一个单词所需要的最少单字符编辑操作次数，常用于测量两个字符串之间差异的字符串度量，可用于起首语分数计算。

2) 问候语

内容上，问候语较为固定，如“你好”“您好”，据此建立标准问候语。

3） 祝颂语

内容上，不做过多限制，表示对留言者的祝愿，或感谢留言者的留言均可。

祝颂语内容自由度较大，但均是表达感谢，祝福之意，可建立标准祝颂语如“感谢”“谢谢”，作为搜索匹配依据。

4） 署名

内容上，标明答复单位，答复者姓名，或答复者工号均可。

因为答复单位不唯一，答复者姓名不确定，署名内容自由度较大，无法做出标准的判断。但署名在答复中位置固定，长度较短，故可依据疑似语句署名长度进行判断。

5） 日期

内容上，只要清楚表明年月日即可。

6） 表示已收到市民留言

最长公共子序列（Longest Common Subsequence, LCS）算法，该算法从给定的两个序列 X 和 Y 中取出尽可能多的一部分字符，按照它们在原序列排列的先后次序排列得到最长公共子序列。

此部分文本长度适中，格式固定，建立关于“已收到”的标准语料库，如“来信收悉，现答复如下：”，将文本内容与标准语料库做比较，得到最长子序列长度值，以此计算相似度。

7） 疑似语句提取

为固定格式，从正文中提取出以上各指标的疑似语句，根据规范答复格式：

- 以标点符号“!”（感叹号），“。”（句号），“？”（问号）为分割依据，可得到从文本中分割出正文。
- 答复第一句话为疑似问候语句
- 正文最后一句话为疑似祝颂语句
- 以正文结束位置为分割，剩余答复内容即是疑似署名语句与疑似祝颂语句

- **算法流程**

1） 起首语

- 疑似起首语与标准起首语语料库比较，计算得到编辑操作次数，取单字符编辑操作次数最小值为 n。
- 因起首语语料文本长度均为 5，若最小编辑次数大于 3，改动过大，即认为不存在起首语，起首语分数置 0。

- 当最小编辑次数小于等于 3 时，如起首语为“同志”，“网友 ASX000”，我们认为这是正常的情况，可计算该文本起首语与标准起首语料库相似度分数。
- 起始语分数：

$$f(n)=\begin{cases}n/5, n\leq 3\\ 0, n<4\end{cases} \tag{6.1}$$

2) 问候语

疑似问候语句搜索匹配，若存在标准问候语给 1 分，不存在记为 0 分。

3) 祝颂语

疑似祝颂语中若存在“感谢”,“祝愿”等词语，则认为存在祝颂语，给 1 分，否则记为 0 分。

4) 署名

设定署名长度阈值为 3，记署名语句长度为 len，对疑似署名语句进行长度检查，若长度大于等于署名长度阈值，则给 1 分。

署名分数：

$$f(len)=\begin{cases}1, len\leq 3\\ 0, len<3\end{cases} \tag{6.2}$$

5) 日期

疑似日期语句若表达日期则给 1 分。

6) 表示已收到市民信件，并明确分割出具体答复正文。

以最长公共子序列（Longest Common Subsequence, LCS）算法将疑似语句与“已收到”标准语料库与答复内容作对比，得到各语料最长子序列长度值(记为 max_length)，取最大值，除以标准语句长度（记为 standard_length），即可得到此项分数。

已收到分数：

$$f(\text{maxL}en)=\max\ \_len/standard\_length \tag{6.3}$$

7) 总分

上述每项评分标准满分一分，总分为 6 分，加和，求得平均分记为该答复完整性分数

6.1.3 答复意见可解释性

给出的答复意见必须可信度高，说服力强，在一定程度上符合常理，满足法律条例规范要求，这样的答复意见才能使发帖者信服，使大众满意，才能解决民众问题，满足大众诉求。

为此，在评价体系中引入可解释性概念。可解释性用于描述答复的合理程度，说服力度。严密而合理的答复内容，可解释性强，使发帖者信服。

- **评价模型**

答复应该是严密的，合理的。通常，有理论支撑的文本会被认为是合理的，比如引用法律条文或明文规定。针对提问情况展开实地勘察或实际调查，也能增加答复的信服力度。

针对政务服务特点，需要理论支撑或实地调查的情况有：

1） 所给信息有错误

如“尊敬的网友：　　您好！您所反映的问题我办已转交相关部门调查处理，因您所提地名有误，我办猜测为石牛江镇，故交由石牛江镇政府调查处理，现将其答复转载如下”

2） 所给信息不全导致无法回答。

如“网友：　　你好！你反映的问题我们进行了调查无法核实，请提供小车车牌号码和联系方式，同时欢迎拨打 12358 价格举报电话和我局联系提供具体的信息，以便我们调查取证。　　2017 年 8 月 25 日”

3） 所提出的问题与实际情况不符。

如“尊敬的网友：你好，经安监、质监部门核实，您举报 B 市泰民米粉厂存在严重安全隐患与事实不符。

以上情况都应给出合理的解释。若并无解释则认为此文本可解释性低。

根据分析给出计算方法：

1） 引用法律条文

答复中引用法律条文的情况有：

第一种情况：用书名号括起文献名称。

第二种情况：为出现书名，但引用了法律条例。如，“根据省厅文件精神”，“让学校根据调档函要求将档案寄至人力资源服务中心”。

根据政务答复文本的书面性较强特点，引用文献时一般使用如下句式：“根据…规定”，“按照…原则”。构建“根据”的同义语料库，构建“规定”的同义语料库，对答复内容进行匹配，得到文献名序列，即为第二种情况的引用法律条文。

两种情况的法律条文序列合并去重统计，即得到引用法律条文数。

统计数据中的各答复引用的法律条文数：

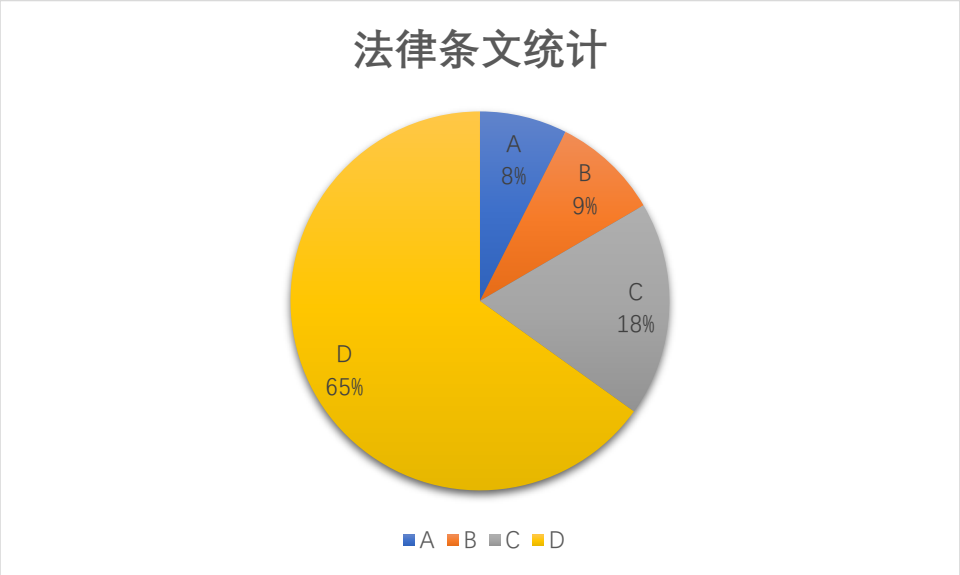


图 33 引用法律条文数统计图

Figure 33 Statistical of the Number of Message which References to Legal Provisions

其中，A 类：法律条文数大于等于三部

B 类：法律条文数为两部

C 类：法律条文数为一部

D 类：未引用法律条文

可知，17%的答复引用了两部及两部以上法律条文，18%的答复引用了一部法律条文，其余 65%未引用法律条文。

2） 实地调查。

构建关于“实地勘察”的语料库，若文本存在能语料库匹配的词语即认为存在实地调查行为。

3） 无法解决问题。

对于信息有误情况，构建关于“错误”的语料库；对于所给信息遗漏情况，构建关于“信息不全”的语料库。若文本与语料库匹配即认为这是无法解决发帖人问题情况。

- **算法流程**

1） 每条答复给基准分 0.4 分

2） 对引用法律条文数作统计

根据答复引用法律条文数目分布情况，制定给分规则如下，其中 n 为引用法律条文数目：

$$f(n)=\begin{cases}0, n=0\\0.15, n=1\\0.3, n\geq 2\end{cases} \tag{6.4}$$

3） 判断是否存在实地勘察行为

存在实地调查行为，给 0.3 分，否则此项给 0 分。

4） 该答复是否无法解决问题

若属于“无法解决发帖者问题”情况，且此时并未识别出理论支撑或给出存在实地调查行为，则认为此答复可解释性低，总分置 0。

6.1.4 答复意见及时性

为及时解决民众问题，满足大众诉求，应在一定时间范围内及时答复发帖者，若拖延时间过长，民众心声得不到回复，认为这是不好的情况。故在评价体系中引入及时性概念，及时性用于描述答复的时效，若答复时间与发帖时间的时间间隔较小，则认为此答复是及时的，在时间上是优秀的。

- **评价模型**

时效性只需要考虑工作日的情况，因为休息日是不上班的，不答复帖子是正常的情况，故在计算答复时间与发帖时间的时间间隔时，需要减去休息日的时间。休息日包括双休日，国家法定节假日。

- **算法流程**

对给分原则进行阐述：

1） 以小时为单位，统计所有帖子的提问时间与答复时间的间隔时间差（不包括双休日）。

2） 求时间差最大值 max，最小值 min；

3） 记 time 为帖子 A 提问时间与答复时间的间隔时间差，则帖子 A 的及时性得分：

$$f(time) = \frac{time}{max - min} \tag{6.5}$$

对时间差计算方法进行阐述：

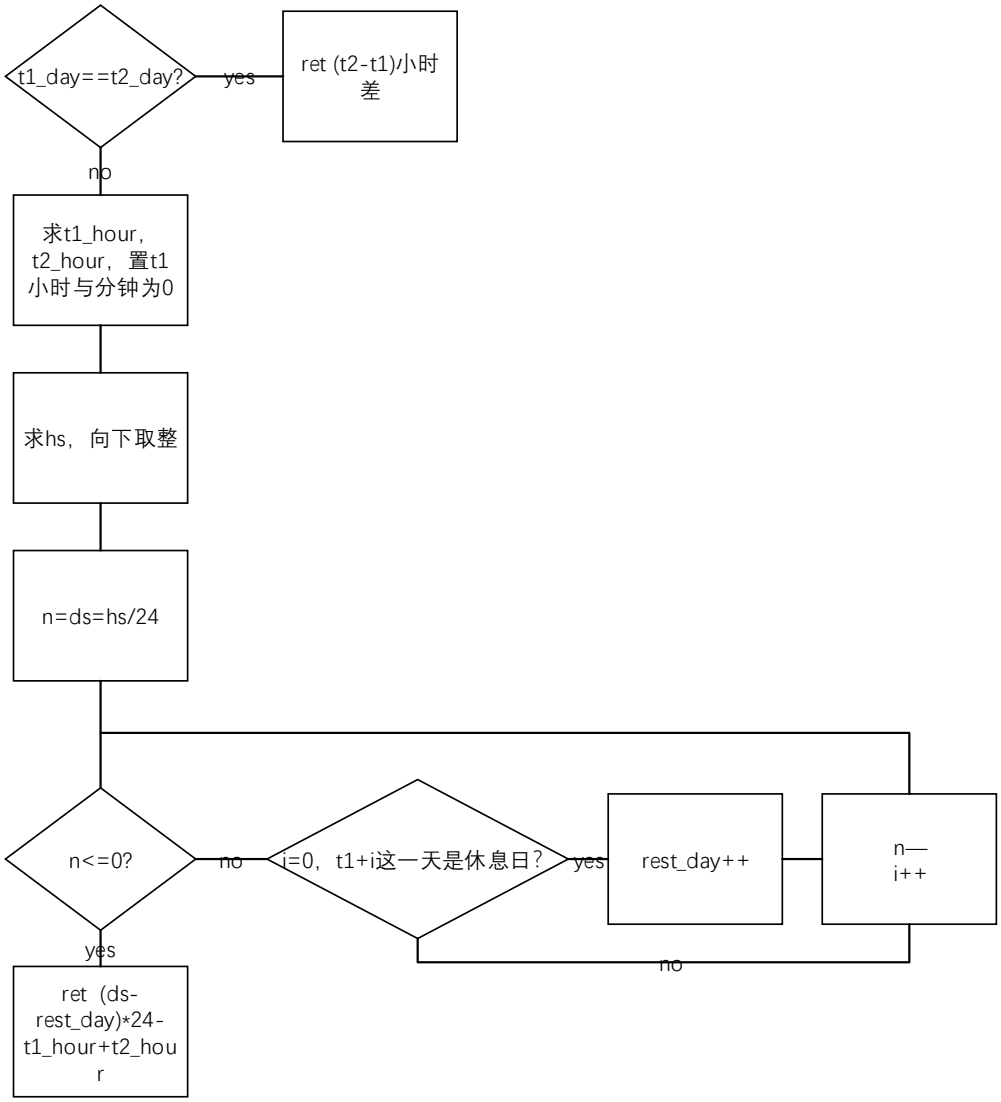


图 34 答复意见与发帖时间时间差计算流程

Figure 34 Calculate Process of Time Difference between Reply and Message time

其中，t1 为发帖时间，t2 为答复时间，hs 为时间间隔的小时数，ds 为时间间隔的天数，rest_day 为休息日天数，t1_day 为 t1 的日期，t2_day 为 t2 的日期，t1_hour 为 t1 的小时数，t2_hour 为 t2 的小时数。

Step1. 若发帖时间（记为 t1）与答复时间（记为 t2）是同一天，计算 t1-t2 的小时差即得到时间差；

Step2. 若 t1 与 t2 不是同一天，将 t1 的小时与分钟清零，计算 t2 与 t1 的间隔天数记为 n（包括 t1，不包括 t2）；

Step3. 逐天判断。记间隔天数 ds 中有 rest_day 天为休息日，根据间隔天数 ds 与 t1，利用 python 中的 chinesecalendar 模块逐天判断是否是节假日，得到 rest_day 的值；

Step4. 计算间隔时间 time（/hour）。记 t1 的小时数为$t2_{hour}$，t2 的小时数为$t2_{hour}$，

给出计算公式如下：

$$time = (ds - rest\_day) * 24 - t1_{hour} + t2_{hour} \quad (6.6)$$

6.2 答复意见类型回归分析

答复长度在一定程度上可以与答复相似性产生联系。单从文本长度而言，长文本相似度高的概率大于短文本的相似度，因为内容更丰富，匹配的概率更高。若答复文本短，而相似度高，则此答复是简明扼要的，可靠性高；答复文本长，而相似度低，那么答复是繁杂冗长的，可靠性低。

因此，将答复意见风格分为两种，一类是“简洁可靠型”，另一类记为“繁杂离题型”。

回归分析是通过建立模型来研究变量之间相互关系的密切程度、结构状态及进行模型预测的一种有效工具。故采用线性回归挖掘出答复相似度和答复长度的隐含关系。

将文本相似度记为 y，答复长度记为 x，做线性回归拟合[30]。

回归结果

对答复文本长度做统计：

- 最大值 7883
- 最小值 3
- 平均值：360
- 大于 2300：16 部，占总文本数 0.56%
- 大于 2000：21 部，占总文本数的 0.7%
- 大于 1000：111 部，占总文本数的 3.94%

答复文本长达不均，需要对答复长度做归一化处理。因为数据集中有少许文本长度过大，且由以上数据可知，文本长度大于 2000 的文本仅占到 0.7%。因为评价模型考虑的是大部分情况，少数文章的特殊性，不会影响到回归的结果。故除去异常的值，仅保留文本长度小于 2300 的数据。

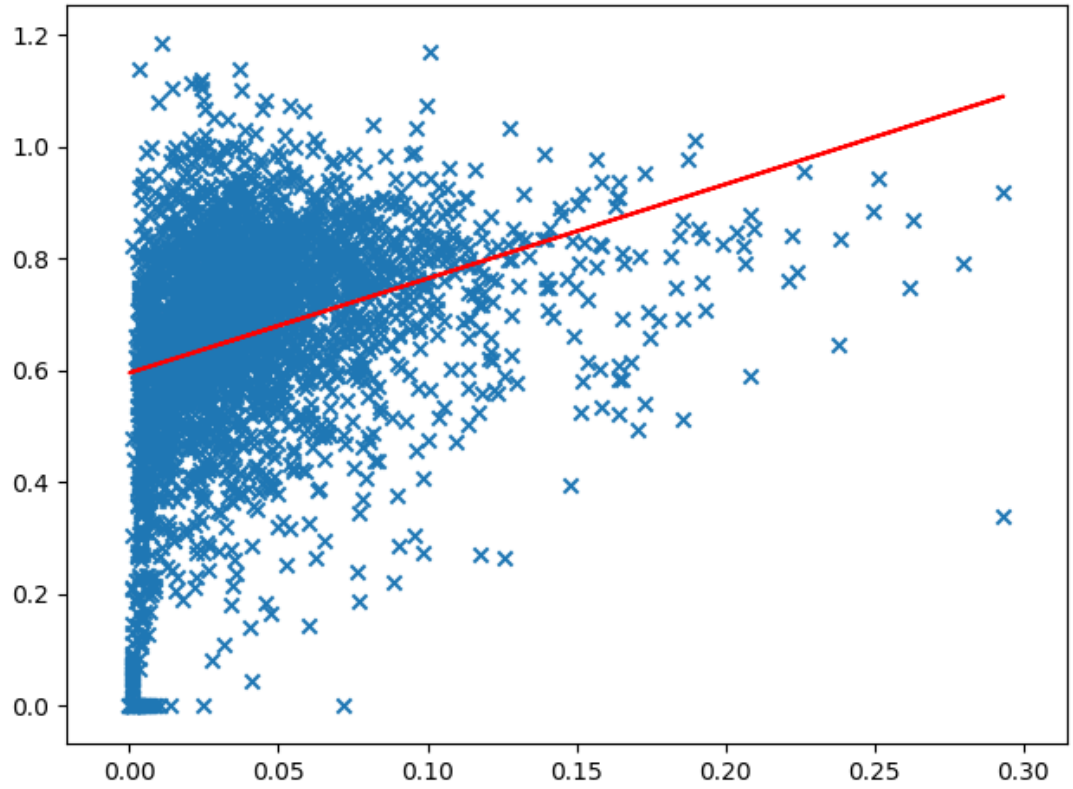


图 35 线性回归结果图

Figure 35 Linear Regression Result

最小二乘法拟合后的直线方程为：

$$y = 1.423x + 0.502 \tag{6.7}$$

根据答复意见相似性与文本长度的关系可知，位于直线上方的答复意见风格是简洁可靠型，位于直线下方的是繁杂离题型。

6.3 答复意见等级聚类分析

由于训练样本是无标签的数据，各项指标仅靠简单的加权求和并不能综合地反映出答复的完备程度，也会忽略答复特征的内在联系，故采用聚类这种无监督学习方式，以完整性，可解释性，及时性三项为特征，对答复文本数据进行分类。聚类的数目定为三类。

表 14 答复意见聚类处理特征

聚类处理的特征		
完整性	可解释性	及时性

聚类的算法有许多，为确定哪种算法更适合本次评价体系，采用基于划分的 k-means，基于密度的 DBSCAN，基于均值迁移的 Mean-shift，以及 Hierarchical Clustering 层次聚

类，并对这几种算法做比较。

表 15 聚类算法

聚类算法			
k-means	DBSCAN	Mean-shift	Hierarchical Clustering

6.3.1 k-means

k-means 算法[31]是典型的基于距离的非层次聚类算法，在最小化误差函数的基础上将数据划分为预定的类数 K，采用距离作为相似性的评价指标，即认为两个对象的距离越近，其相似度就越大。具体算法过程如下：

Step1. N 个评价特征数据中随机选取 K 个样本作为初始的聚类中心；

Step2. 分别计算每个样本到各聚类中心的距离，将对象分配到最近的类簇中；

Step3. 所有对象分配到所属类簇后，重新计算 K 个类簇的聚类中心；

Step4. 与步骤三种计算得到的 K 个聚类中心比较，如果聚类中心发生变化，转 Step2，否则 Step5；

Step5. 当聚类中心不发生变化时停止计算，并输出聚类结果。

特征数据计算训练后，根据聚类结果绘制分布图如下：

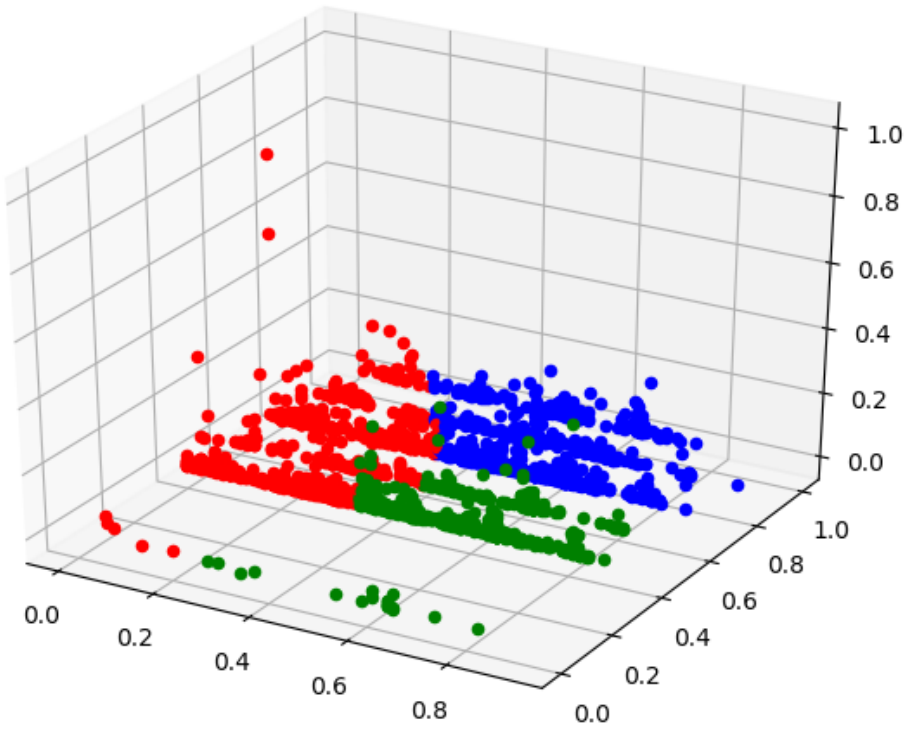


图 36 k-means 聚类分布

Figure 36 K-means Clustering Distribution

为特征数据定义三个等级，等级一二三分别对应优良中，则图中蓝色为等级一，绿色为等级二，红色为等级三。

6.3.2 DBSCAN

DBSCAN 是一种著名的密度聚类算法，它基于一组“邻域”参数来刻画样本分布的紧密程度。其将簇定义为：由密度可达关系导出的最大密度相连样本集合。具体算法过程如下：

Step1. 将所有数据对象标记为核心点、边界点或噪声点；

Step2. 删除噪声点；

Step3. 为距离在 Eps=0.5 之内的所有核心点之间赋予一条边；

Step4. 每组联通的核心点形成一个簇；

Step5. 将每个边界点分配到一个与之关联的核心点的簇中。

特征数据计算训练后，根据聚类结果绘制分布图如下：

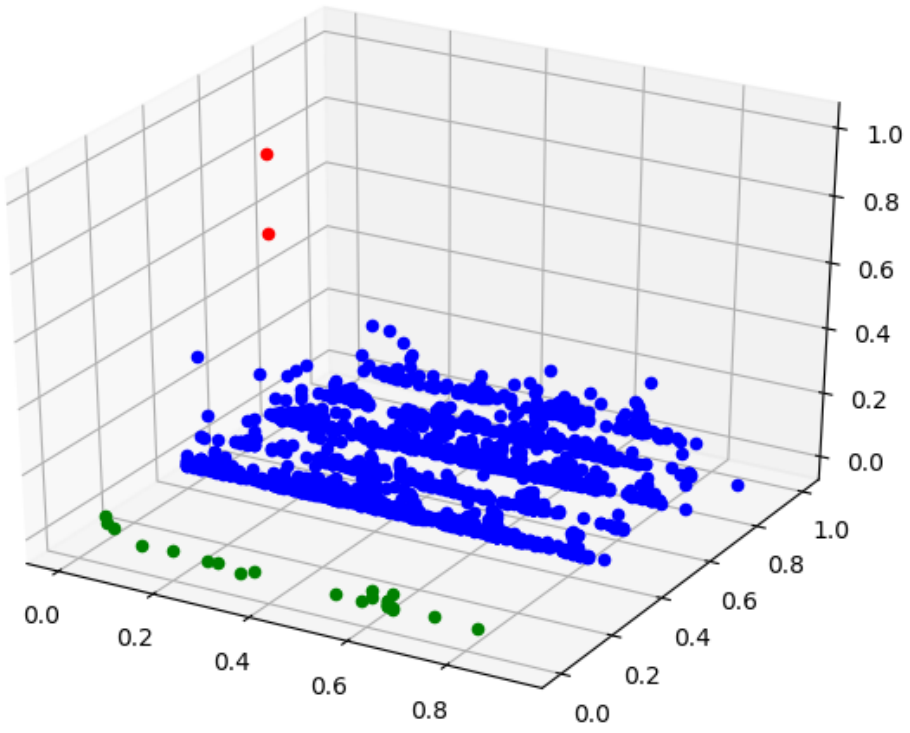


图 37 DBSCAN 聚类分布

Figure 37 DBSCAN Clustering Distribution

为特征数据定义三个等级，等级一二三分别对应优良中，则图中蓝色为等级一，绿色为等级二，红色为等级三。

6.3.3 Mean-shift

均值漂移(Mean-Shift)算法是一种无参密度估计算法或称核密度估计算法,Mean-shift是一个向量，它的方向指向当前点上概率密度梯度的方向。在聚类中，该算法完全依靠特征空间中的样本点进行分析，不需要任何先验知识，数据集中的每一点都可以作为初始点，它能对任何维度、任何分布的采样点进行快速聚类，迭代效率高。具体算法过程如下：

Step1. 在未被标记的数据点中随机选择一个点作为起始中心点 center;

Step2. 找出以 center 为中心半径，半径为 radius 的区域中出现的数据点，记为集合 M，认为这些点同属于一个聚类 C，同时将这些点属于此类概率加一；

Step3. 以 center 为中心点，计算从 center 开始到集合 M 中每个元素的向量并求和，得到向量 shift;

Step4. 重复步骤二，步骤三，步骤四，直到 shift 迭代到收敛，记下 center 值；

Step5. 若收敛时当前簇 C 的 center 与其他已存在的簇 C2 中心的距离小于阈值，则将 C 与 C2 合并。否则，将 C 作为新的聚类。

Step6. 重复步骤一、二、三、四、五，直到所有的点都被标记访问；

Step7.根据每个类对每个点的访问频率，取访问频率值最大的类，作为当前点集的所属类。

特征数据计算训练后，根据聚类结果绘制分布图如下：

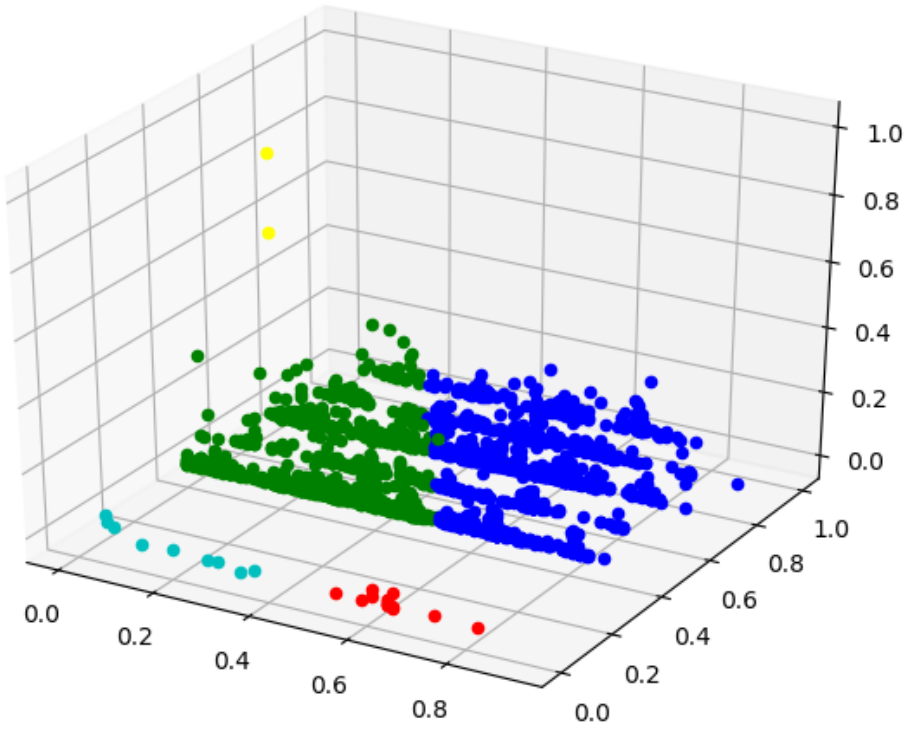


图 38 mean_shift 聚类分布

Figure 38 Mean_Shift Clustering Distribution

为特征数据定义五个等级，等级一二三四五分别对应优秀，较优秀，良好，较良好，中，则图中蓝色为等级一，绿色为等级二，红色为等级三，湖绿色为等级四，黄色为等级五。

6.3.4 Hierarchical Clustering

层次聚类的合并算法通过计算两类数据点间的相似性，对所有数据点中最为相似的两个数据点进行组合，并反复迭代这一过程，最终生成聚类树。具体算法过程如下：

Step1. 假设每个样本为一类，计算每个类的距离；

Step2. 将距离最近的两类合为一新类；

Step3. 计算新类与各个旧类之间的相似度；

Step4. 循环重复步骤二和步骤三，直到所有样本点都归为一类

特征数据计算训练后，根据聚类结果绘制分布图如下：

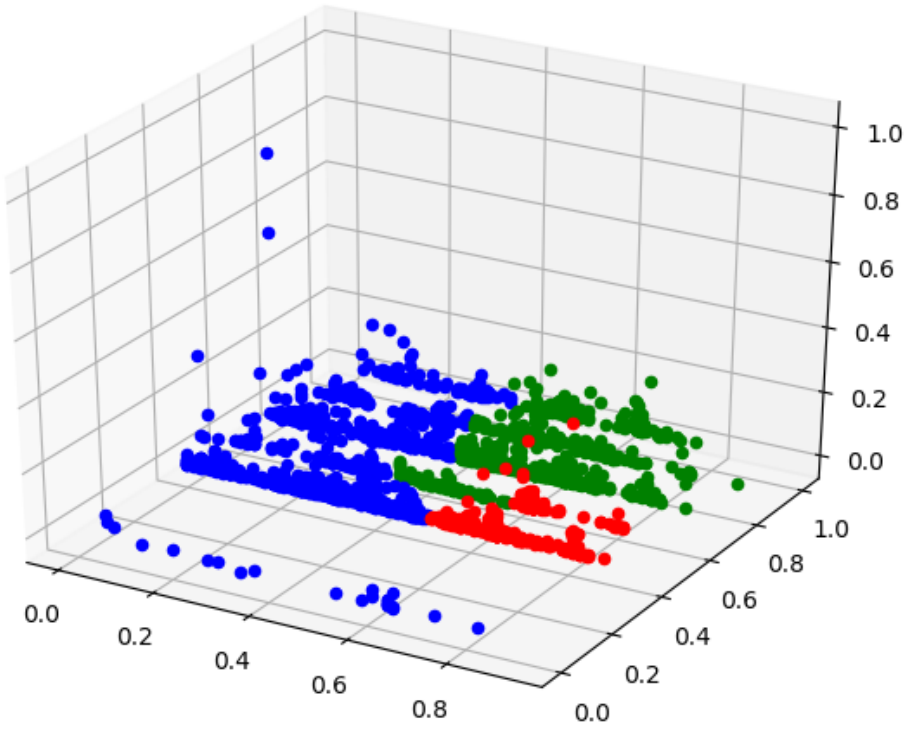


图 39 Hierarchical Clustering 聚类分析

Figure 39 Hierarchical Clustering Clustering Distribution

为特征数据定义三个等级，等级一二三分别对应优良中，则图中绿色为等级一，红色为等级二，蓝色为等级三。

6.3.5 聚类效果对比分析

采用 Calinski-Harabasz 指数和轮廓指数（Silhouette Coefficient）这两种指标来对聚类效果进行评价。

- Calinski-Harabasz 指数

评价分数计算公式如下：

$$s(k) = \frac{tr(B_k)m - k}{tr(W_k)k - 1} \tag{6.8}$$

其中：m 为训练样本数；k 为类别数；B_k为类别之间的协方差矩阵；W_k为类别内部数据的协方差矩阵；tr 为矩阵的迹。

即类别内部数据的协方差越小越好，类别之间的协方差越大越好。CH 的目的是，用尽量少的类别聚类尽量多的样本，同时获得较好的聚类效果。

- **轮廓系数**

轮廓系数结合了聚类的凝聚度（Cohesion）和分离度（Separation），用于评估聚类的效果。对于单个样本，设 a 是与它同类别中其他样本的平均距离，b 是与它聚类最近的不同类别中样本的平均距离，则轮廓系数为：

$$S = \frac{b - a}{\max(a, b)} \tag{6.9}$$

对于一个样本集合，它的轮廓系数是所有样本轮廓系数的平均值，即平均轮廓系数。

平均轮廓系数的取值范围是[-1，1]，同类别样本距离越近且不同类别样本距离越远，分数越高，聚类效果越好。

- **聚类效果分析**

表 16 聚类效果评价表

Table 14 Evaluation of Clustering Effect

聚类名称	Calinski-Harabasz 指数	轮廓系数
k_means	2355.65	0.43823
DBSCAN	65.23	0.44938
mean_shift（5 个簇）	489.58	0.36499
HC	1768.03	0.38856

从 Calinski-Harabasz 指数来看，k-means 聚类明显大于其余聚类算法，说明其类别内部数据的协方差更小，且类别之间协方差更大，使得答复优良程度的分类更加清晰。另外，从轮廓系数来说，虽然 DBCAN 算法比 k-means 算法稍大，但由聚类图可知，DBSCAN 算法将绝大部分数据归为一类，这算是不理想且不符合预期的。除去 DBSCAN，k-means 算法的轮廓系数大于其他聚类算法，说明其同类别样本距离更近且不同类别样本距离更远，聚类结果更为准确。

因此，由这两个指数的对比可知，对于答复意见，k-means 聚类算法较为理想。

6.4 结果分析

综合考虑答复意见特征与答复类型两个指标，将答复优良等级分为六个等级。如下表所

示：

表 17 答复意见评价等级表

Table 15 Reply Scale

答复风格	答复意见优良等级		
	等级一	等级二	等级三
简洁可靠型	优秀	较优秀	良好
繁杂离题型	较良好	较一般	一般

对 2816 则答复进行预处理，提取答复意见特征，再对答复风格进行回归分析，得到答复风格回归直线。其中：回归直线上的为简洁可靠型，有 1456 则，占总文本的 52%；回归直线下方的为繁杂离题型，有 1360 则，占总文本的 48%。

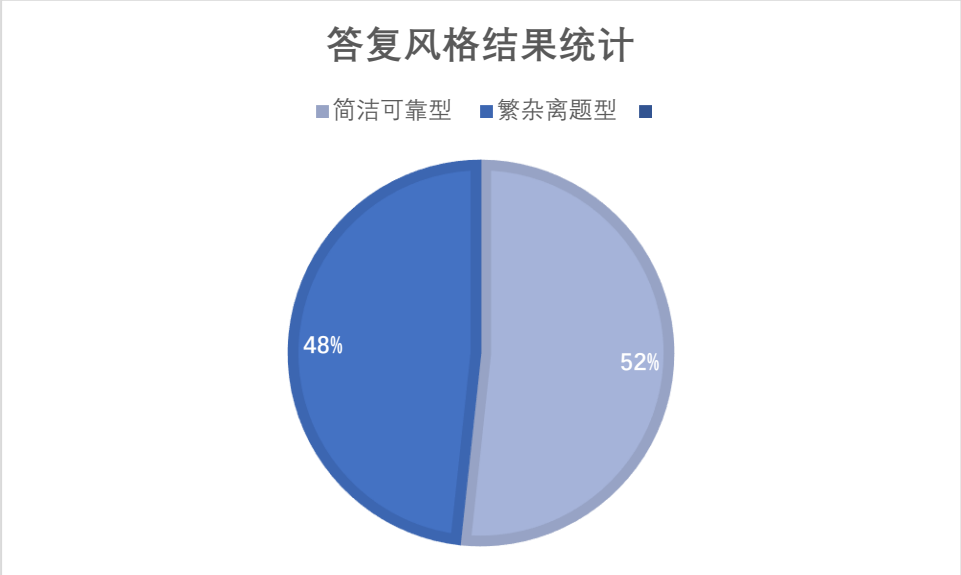


图 40 答复风格统计图

Fig.40 Reply style statistics

有图表可知，繁杂离题型答复几乎占据了总文本数的一半，政务工作者答复时应尽量简洁凝练，为民众给出较为核心的解决办法。

采用 k-means 算法进行分析，得到答复意见优良程度的三个等级。其中：

- 第一等级有答复 1066 则，占总答复的 37.855%
- 第二等级有答复 915 则，占总答复的 32.493%
- 第三等级有答复 833 则，占总答复的 29.581%

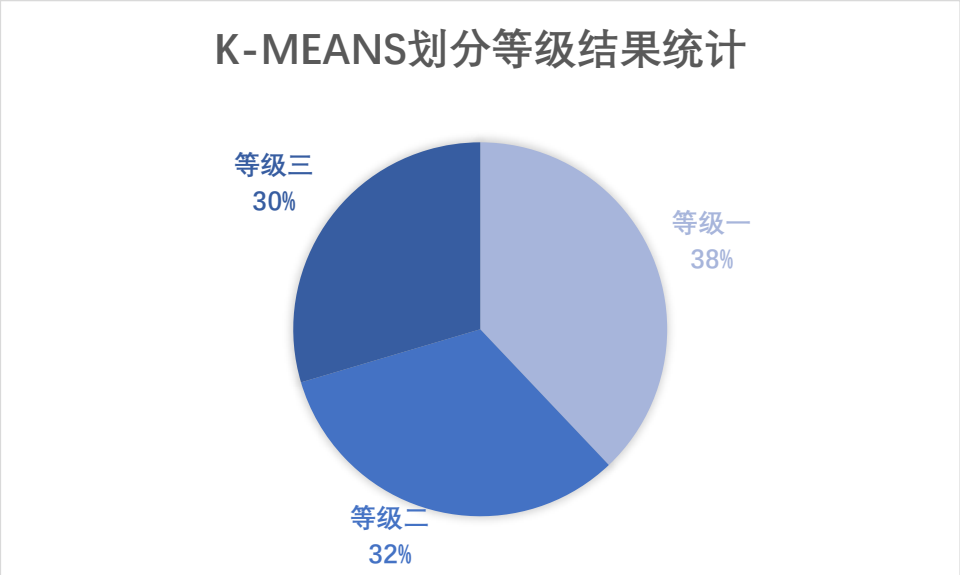


图 41 k-means 划分等级结果统计图

Fig.41 K-means Classification Results Statistical Graph

由图表可知，答复意见等级分布均匀，但仍有近 1/3 的答复文本的是较不规范的，政务工作人员应重视此问题，对答复质量做出改进。

综合分析以上两个评级指标，分析 2816 则答复的优良程度。将简洁可靠型+等级一评价为优秀；简洁可靠型+等级二评价为较优秀；简洁可靠型+等级三评价为良好；繁杂离题型+等级一评价为较良好；繁杂离题型+等级二评价为较一般；繁杂离题型+等级三评价为一般。评价结果如下图所示：

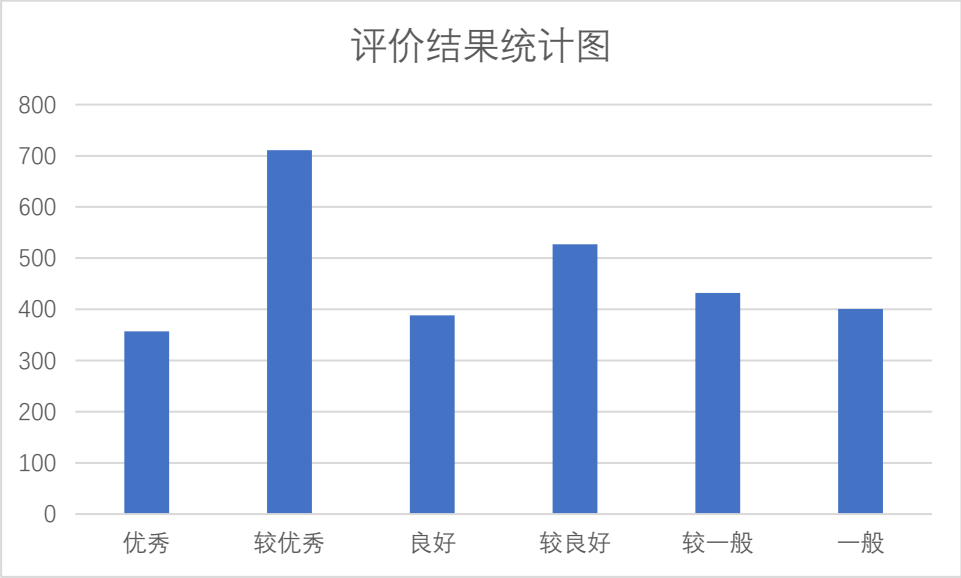


图 42 评价结果统计图

Fig.42 Statistical Graph of Evaluation Results

1) 优秀：357 则，占总文本数 12.677%

2) 较优秀：711 则，占总文本数 25.249%

3） 良好：388 则，占总文本数 13.778%

4） 较良好：527 则，占总文本数 18.714%

5） 较一般：432 则，占总文本数 15.34%

6） 一般：401 则，占总文本数 14.24%

综上，可见评价结果是均匀的，良好的，故评价体系是可靠的。政务工作人员应加强对答复质量的重视程度，给出更行之有效，更规范，更优秀的答复意见，完善政务服务工作。

第七章 总结与期望

电子政务作为信息化时代政府服务方式转变途径之一，在政府的日常工作中发挥着重要作用。政务问答作为政府与民众对话交流的有效手段，得到了越来越多的肯定，成为政务服务的重要组成部分。而电子政务文本数量的加剧，内容的繁杂不一，以及对及时答复的要求，使得工作量不断增大。因此，使用智能方式对电子政务信息管理至关重要。

本文的目的是根据政务问答文本数据提出一套管理方法，便于提高政务处理效率与服务水平。针对问题一，对各种文本表示方法与机器学习算法进行组合实验，并对深度学习模型进行分类实验。在此基础上，分别对模型提出优化，对机器学习应用 CPSO 算法和 IAPSO 算法，对深度学习模型提出了一种带类别权重的卷积神经网络模型；针对问题二，对基于 Singlepass 算法和 LDA 模型的话题划分方法作比较，并提出热度计算公式，发现话题热点；针对问题三，定义答复意见文本特征，分别用回归分析和聚类分析对答复意见进行评价，并总结为六类。

在本次文本处理的过程中，对同一问题，采用多种算法进行分析并对比，整个过程有理有据且条理清晰。但是由于时间有限，水平有限，有一些值得研究和改进的地方，比如问题三，可针对情感极性，语法正确率提出新特征，会有更好的评价体系。在今后的学习中，我们将尝试使用全新的解决思路，对目标问题提出更加全面有效的解决方案。

参考文献

[1] 李质轩. 融合上下文信息的汉语分词方法研究[D].北京交通大学,2018.

[2] 曹勇刚,曹羽中,金茂忠,刘超.面向信息检索的自适应中文分词系统[J].软件学报,2006(03):356-363.

[3] 张超. 一种词性标注 LDA 模型的文本分类方法研究[D].华中师范大学,2015.

[4] 任刚. 面向学科相关性分析的文本关联规则挖掘技术研究[D].中南大学,2011.

[5] 罗杰,陈力,夏德麟,王凯.基于新的关键词提取方法的快速文本分类系统[J].计算机应用研究,2006(04):32-34.

[6] 周钦强,孙炳达,王义.文本自动分类系统文本预处理方法的研究[J].计算机应用研究,2005(02):85-86.

[7] Pang, Bo, and Lillian Lee. "Opinion mining and sentiment analysis." Foundations and Trends® in Information Retrieval 2.1-2 (2008): 1-135.

[8] Song, Fengxi, Shuhai Liu, and Jingyu Yang. "A comparative study on text representation schemes in text categorization." Pattern analysis and applications 8.1-2 (2005): 199-209.

[9] Porter M F . An algorithm for suffix stripping[J]. Program, 1980, 14(3):130-137. Hotho, Andreas, Andreas Nürnberger, and Gerhard Paaß. "A brief survey of text mining." Ldv Forum, 2005:19-62

[10]Lancaster, Frederick Wilfrid, and Emily Gallup. Information retrieval on-line. No. Book. 1973., Salton, Gerard, Edward A. Fox, and Harry Wu. "Extended Boolean information retrieval." Communications of the ACM 26.11 (1983): 1022-1036.

[11]Salton, Gerard, Anita Wong, and Chung-Shu Yang. "A vector space model for automatic indexing." Communications of the ACM 18.11 (1975): 613-620.

[12]Chowdhury, Gobinda G. Introduction to modern information retrieval. Facet publishing, 2010.)

[13]Wei, Chih-Ping, Christopher C. Yang, and Chia-Min Lin. "A latent semantic indexing-based approach to multilingual document clustering." Decision Support Systems 45.3 (2008): 606-620.

[14]HOTHO A, MAEDCHE A, STAAB S. Ontology-based text document clustering[J].KI, 2002, 16(4):48-54.

[15]CAVNAR,W. Using An N-Gram-Based Document Representation With A Vector Processing Retrieval Model.[J]. proc of trec, 1994:269-277.

[16]Le Q V , Mikolov T . Distributed Representations of Sentences and Documents[J]. 2014.

[17]Breiman, L, Breiman, Leo, Cutler, Raymond A. Random Forests Machine Learning[J]. journal of clinical microbiology, 2001, 2:199-228.

[18]CHEN, Tianqi; GUESTRIN, Carlos. XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016. p. 785-794.

[19]Shi Y,Eberhart R. A modified particle swarm optimizer[C]. IEEE International Conference on Evolutionary Computation Proceedings，IEEE World Congress on Computational Intelligence，IEEE， 1998: 69-73.
[20]朱海梅, and 吴永萍. A PSO algorithm with high speed convergence.[J] 控制与决策,025.001(2010):20-24,30. 吕振肃, 侯志荣, Lu Z S . 自适应变异的粒子群优化算法[J]. Acta Electronica Sinica, 2004, 32(3):416-420.
[21]Ying Song，Chen Zengqiang，Yuan Zhuzhi. New chaotic PSO-based neural network predictive control for nonlinear process[J].IEEE Transactions on Neural Networks，2007，18(2)：595-601. Zhou Keliang，Qin Jieqiong. PID controller parameters tuning of main steam temperature based on chaotic particle swarm optimization[C]//IEEE International Conference on Computer Science and Automation Engineering(CSAE)，2011：647-650.
[22]唐贤伦， et al.一种基于多目标混沌 PSO 的机器人足球防守策略. 系统仿真学报,01(2014):55-59+65.
[23]CHENG, Min-Yuan; HUANG, Kuo-Yu; CHEN, Hung-Ming. K-means particle swarm optimization with embedded chaotic search for solving multidimensional problems. Applied Mathematics and Computation, 2012, 219.6: 3091-3099.
[24]Xiang, Tao, Xiaofeng Liao, and Kwok-wo Wong. "An improved particle swarm optimization algorithm combined with piecewise linear chaotic map." Applied Mathematics and Computation 190.2 (2007): 1637-1645.
[25]Xiang, Tao, Xiaofeng Liao, and Kwok-wo Wong. "An improved particle swarm optimization algorithm combined with piecewise linear chaotic map." Applied Mathematics and Computation 190.2 (2007): 1637-1645.
[26]路玉君. 基于 RNN 的陆空通话语义描述与度量方法[D].中国民航大学,2017.11,18
[27]Blei D M , Ng A Y , Jordan M I , et al. Latent Dirichlet Allocation[J]. Journal of Machine Learning Research, 2003, 3:993-1022.
[28]李文峰. 基于主题模型的用户建模研究[D].北京邮电大学,2013:16-18,25-27.
[29]杨莉,王敏,程宇.基于 LDA 和 XGBoost 模型的环境公共服务微博情感分析[J].南京邮电大学学报(社会科学版),2019,21(06):23-39.
[30]Timofey Samsonov, Olga Yakimova. Regression modeling of reduction in spatial accuracy and detail for multiple geometric line simplification procedures. International Journal of Cartography. 10.1080/23729333.2019.1615745
[31]任远航. 面向大数据的 K-means 算法综述[J/OL]. 计算机应用研究.