ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 41,No. 3,May 2026,pp. 795－813
DOI:10. 16337/j. 1004-9037. 2026. 03. 013

http://sjcj. nuaa. edu. cn
E-mail:sjcj@nuaa. edu. cn
Tel/Fax:＋86-025-84892742

融合文本特征与词频隐因子的线性注意力文本分类

苏　湛,张　旭,艾　均,徐温果

(上海理工大学光电信息与计算机工程学院,上海 200093)

摘　要: 在文本分类任务中,有效地提取文本特征并提高计算效率是关键问题,但传统方法难以同时兼顾特征丰富性和计算效率。针对这一问题,本文提出了一种融合文本特征与词频隐因子的文本分类(Linear attention text classification by combining text features and word frequency implicit factors,LTTW)模型,并引入线性注意力机制来捕捉文本关键特征。模型通过非负矩阵分解(Non-negative matrix factorization,NMF)从词频矩阵中提取词频隐因子,以捕捉潜在语义信息;同时,利用预训练模型提取文本语义特征,并与词频隐因子融合,构建更为丰富的文本表示。在此基础上,采用线性注意力机制,有效捕捉全局依赖关系并提高长文本序列的处理效率。本文在公开数据集上进行了实验验证,结果显示,所提出的模型在准确性和计算效率上均优于现有主流方法,尤其在处理长序列数据时表现出显著的效率优势。研究表明,词频隐因子的引入补充了预训练模型在语义特征提取方面的不足,线性注意力机制能够在有效捕捉文本关键特征的同时提高序列处理的效率,有效提升了文本分类的效果和效率。

关键词: 文本分类;线性注意力机制;隐因子;文本特征;矩阵分解

中图分类号: TP183　　**文献标志码:** A

引用格式: 苏湛,张旭,艾均,等.融合文本特征与词频隐因子的线性注意力文本分类[J].数据采集与处理,2026,41(3):795-813. SU Zhan, ZHANG Xu, AI Jun, et al. Linear attention text classification by combining text features and word frequency implicit factors[J]. Journal of Data Acquisition and Processing, 2026, 41(3):795-813.

引　　言

文本分类的目的是将文本数据自动分配到特定的标签或类别中,是自然语言处理领域中应用最广泛且最重要的技术之一。目前,文本分类已被广泛应用于与日常生活密切相关的多个领域,如新闻分类[1]、主题标记[2]、问答系统[3]以及垃圾邮件检测[4]等。由此可见文本分类任务是一项关键的基础技术。同时随着研究的深入,文本分类领域内的关键技术也在不断改进,进一步提高了分类的准确性和应用的广泛性。

传统的机器学习方法通过词语的语义获取特征,从而实现文本的自动化分类,大幅提升了文本处理的效率。这类方法在分类过程中通常包含3个主要步骤[5]:文本预处理、特征提取以及分类计算。尽管传统机器学习方法在分类的准确性和稳定性方面表现优异,但其在实际应用中仍面临一些显著局限[6]。首先,这些方法高度依赖于耗时且成本高昂的特征工程。其次,由于其强烈依赖领域知识,传统方法在面对新的分类任务时的可扩展性和有效性常常受到限制。此外,这些方法往往忽略了文本中的序列信息、上下文信息以及单词本身的深层语义,这与人类对句子的理解方式并不完全一致[7]。由于语

基金项目: 国家自然科学基金(61803264)。
收稿日期: 2024-12-04;**修订日期:** 2025-02-18

言本身具有笼统性、复杂性与多义性等特点,使得对文本中蕴含的语义信息进行有效挖掘成为一项极具挑战性的任务。随着文本分类技术的发展,文本分类发展呈现以下趋势:

(1)深度模型的应用。近年来,基于深度学习的方法在文本分类任务中得到了广泛应用。与传统方法依赖人工设计规则和特征不同,深度学习通过构建深层非线性网络结构,能够自动从数据中提取文本的核心特征,有效捕捉文本的深层语义表征[8]。此外,深度学习模型能够处理多种类型的特征,并将其映射到统一的向量空间,从而生成统一的特征表示[9]。这一特性使得模型能够融合多种数据来源,缓解语义歧义和信息不足等问题,从而实现更加全面和精准的分类效果。随着Transformer模型的崛起,BERT(Bidirectional encoder representation from transformers)[10]、ERNIE(Enhanced representation through knowledge integration)[11]和GPT(Generative pre-trained transformer)[12]等一系列大规模预训练语言模型相继被提出。深度学习模型无需人工标注,能够从海量语料中学习通用的语言表示,显著提升了下游任务的表现,显著提高了文本分类的准确率。

(2)可扩展性。随着大规模数据集的增长以及对实时处理需求的提升,模型的可扩展能力变得至关重要。为应对这一挑战,研究逐渐转向轻量化模型和高效的注意力机制,以降低计算复杂度并提高模型在长序列文本上的表现。Jiao等[13]探讨了模型压缩技术,Sheng等[14]探讨了线性注意力机制在文本分类中的应用。提升可扩展性能够在保持准确性的同时显著优化计算资源的使用,特别是在处理海量数据和多语言、多领域任务时,具有广泛应用前景。

(3)长文本分类。文本分类模型的另一个重要发展趋势是针对长文本的处理能力。传统深度学习模型,如LSTM(Long short term memory)[15]、Transformer[16]在处理长文本时面临计算资源瓶颈,因此如何高效处理长文本成为研究热点。Beltagy等[17]提出了一种能够高效处理长文档的Transformer模型,通过稀疏注意力机制大幅降低了计算复杂度,并在多个长文本分类任务中取得了领先成绩。这些模型的提出可以有效缓解深度学习模型在面临长文本序列时的瓶颈,提升长文本分类的准确率和效率。

文本分类任务在快速发展的同时也面临着诸多挑战:(1)预训练模型的不足[18]。对于预训练模型,如BERT等虽然在全局语义建模上表现出色,但在识别和利用局部词频等关键特征时,特征丰富度可能不足,无法充分反映文本分类任务中的一些重要信息,造成准确率不高的结果;(2)训练成本高昂[19]。预训练模型和传统注意力机制的计算复杂度高,需要大量的计算资源,且在某些应用场景中,数据更新频繁,导致模型需要频繁微调,大大增加了训练成本[20]。(3)模型的可解释性[21]。随着深度学习模型的复杂性增加,其内部工作机制变得越来越难以理解。如何提升文本分类模型的可解释性,使得用户能够理解模型的预测理由,同样是一个挑战。

针对预训练模型的不足和训练成本高昂的挑战,本文提出了融合文本特征与词频隐因子的线性注意力文本分类(Linear attention text classification by combining text features and word frequency implicit factors, LTTW)模型。针对预训练模型更新迭代快等挑战,本文提出了通过对数据集的文本和标签信息构建词频矩阵,然后对矩阵进行非负矩阵分解,增加词频的隐因子来丰富文本内容。针对传统注意力机制训练效率低成本高的挑战,在通过引入线性注意力机制处理特征来捕捉文本关键特征的同时提高模型对长序列处理的效率,避免了增加额外特征造成效率下降的问题。

1 相关工作

矩阵分解是一种将矩阵拆解为多个矩阵相乘的过程[22]。常用于数据降维和特征提取的技术,通过将大型数据矩阵分解为多个较小的矩阵,简化数据表示并降低计算复杂度。在自然语言处理领域,非负矩阵分解是其中一种常见的变体[23],尤其在文本分类任务中得到了广泛应用。通过矩阵分解技术,高维文本数据能够被映射到低维潜在空间,同时保留文本的主要特征[24]。这一方法在应对数据稀疏性和高维性问题时表现突出,有助于提高分类器的性能。Mao等[25]的研究指出,非负矩阵分解可以有效用于文本特

征提取和选择,通过去除冗余特征和噪声来提高分类精度。在本文中,非负矩阵分解被用于对词频矩阵进行分解,这样不仅可以降低词频矩阵的维度,同时基矩阵的隐因子也可用于丰富文本特征。

BERT是由Google于2018年提出的一种预训练语言模型,基于Transformer架构。BERT的核心创新在于其双向编码器结构,该结构使模型能够同时关注文本的前后文信息,而非像传统语言模型那样只能单向处理文本[10]。这种双向性赋予BERT更强的能力来捕捉复杂的语义关系和上下文依赖,从而在各种自然语言处理任务中展现出卓越的表现。本文将BERT预训练模型进行微调来获取文本特征,同时学习上下文之间的深层次关系,大大提升其对文本语义的理解能力。

注意力机制是一种深度学习技术,旨在使模型能够在处理数据时聚焦于最相关的信息片段,而忽略不重要的信息[16]。通过分配不同的注意力权重,注意力机制能够突出不同特征的重要性,特别适用于自然语言处理任务,如机器翻译、文本摘要和情感分析。近年来,研究人员不断致力于提升注意力机制的计算效率。Zhai等[26]提出的无关注变换器通过使用一系列线性变换和替代计算,取代了传统的复杂注意力计算。这一方法不仅显著降低了计算成本,还能保持模型的表达能力,尤其在长序列建模任务中表现出色。该架构在某些任务中的表现与传统Transformer相当,但在计算效率方面具有显著优势。在本文中线性注意力机制被用于处理融合后的词频隐因子和预训练模型特征,达到捕捉文本关键特征和提升计算效率的目的。

本文提出的方法融合了词频矩阵分解的隐因子和预训练模型特征,然后使用注意力机制对特征进行处理,旨在通过丰富文本特征降低计算复杂度,达到提升文本分类的效率和准确率的目的。

2　本文方法

2.1　问题描述

文本分类的目标可以描述为:给定数据集$D_{\text{train}}=\{(t_i)\}_{i=1}^{U}$,其中$t_i$为文本,$C$为标签且$C=\left\{c_i\in\{0,1,\cdots,p\}_{i=1}^{U}\right\}$,$p$为文本类别的总数,$U$为数据集的样本总数。文本分类任务的目标是学习一种预测模型,能够将一个新的未标记的文本分类到现有标签中。本文用到的符号如表1所示。

表1　符号说明

Table 1　Symbol specification

符号	说明
D_{train}	训练集数据
W_n	单词n
C_p	标签
$W_{\text{Top\_}k}$	词频第k高的单词
$H_{\text{Top\_}k}$	词频第k高的单词特征
$W_{\text{Bottom\_s}}$	词频前s低的单词
$H_{\text{Bottom\_s}}$	词频前s低的单词特征
$V_{n\cdot p}$	词频矩阵
$H_{p\cdot m}$	隐因子矩阵
$W_{n\cdot p}$	矩阵分解系数
H_{iw}	文本的词频隐因子特征
H_{BERT}	BERT特征维度
y_{iw}	词频隐因子维度
$\boldsymbol{y}_{\text{BERT}}$	BERT预训练模型特征
H_V	融合后的文本特征
P	标签概率
Y	文本全局特征

2.2　模型整体框架

模型由4个关键模块组成:词频隐因子模块、预训练特征模块、线性注意力模块和分类模块,模型结构如图1所示。先使用数据集的文本和标签构建词频矩阵,然后根据词频从分解的矩阵内获取特征,然后使用预训练模型获取文本的预训练特征,将词频隐因子和预训练模型特征进行融合后使用线性注意力机制进行处理,最后将处理后的特征输入分类模块,最后输出标签。

本文对数据集做以下处理:使用随机抽取的方式获取实验数据,其中分为训练集D_{train}和测试集D_{test}。为避免数据泄露,本文的特征提取和模型训练的数据仅来自于原始训练集。同时本文对训练集进行进一步随机划分为10份,进行折十交叉验证,其中按照8∶1∶1的比例划分为训练集、测试集和验证集。

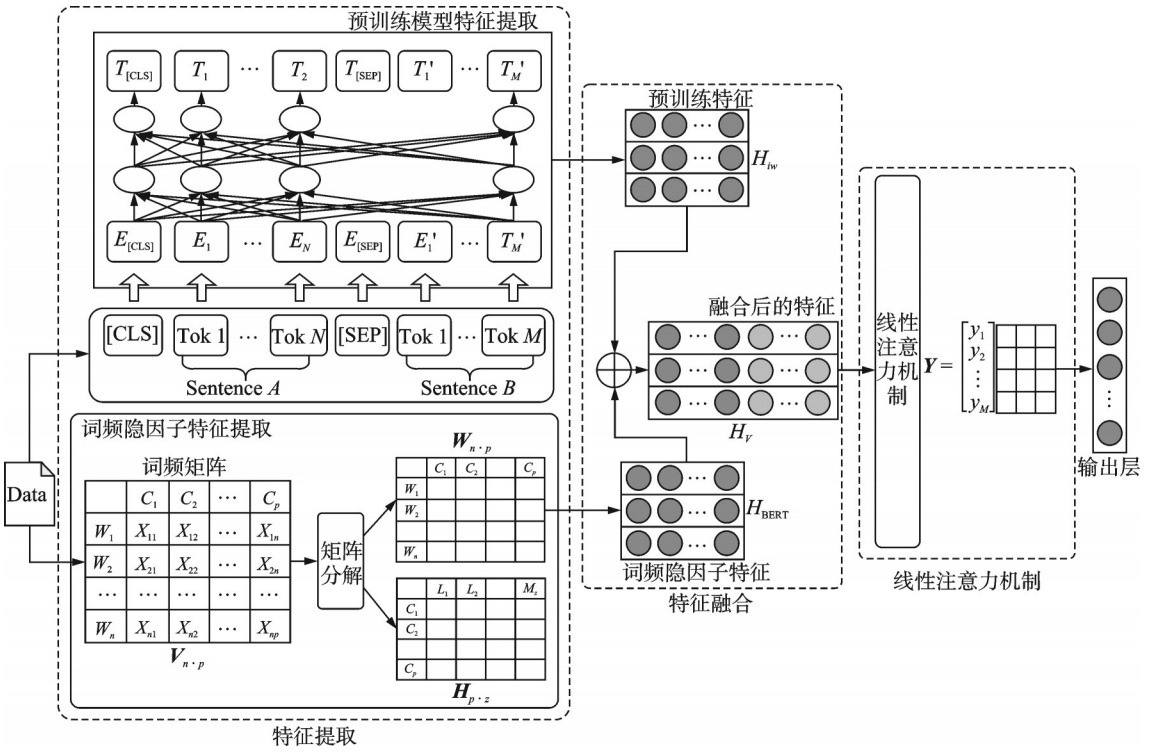

图1 模型整体结构图

Fig.1 Overall structure of the model

词频隐因子模块的主要任务是获取文本的词频隐因子。首先提取数据集所有不重复的单词和标签构建词频矩阵 $\boldsymbol{V}$,然后对词频矩阵进行非负矩阵分解为矩阵 $\boldsymbol{H}$ 和 $\boldsymbol{W}$。接下来对每一条文本的单词按词频从高到低排序,在 $\boldsymbol{H}$ 矩阵内按照词频获取前 k 和后 s 个单词的词频隐因子,最后按照词频顺序将特征拼接后得到文本的词频隐因子。

预训练特征模块的主要任务是使用预训练模型特征提取文本的特征。首先对数据集进行处理获取数据集的文本,然后使用BERT预训练模型提取文本特征。

线性注意力模块的主要任务是将词频隐因子和预训练模型的特征进行融合,然后使用线性注意力机制对特征进行处理。首先将词频隐因子和预训练模型特征进行拼接融合处理,然后使用线性注意力机制对融合后的特征进行处理,捕捉上下文依赖关系的同时降低训练成本。

分类模块的主要任务是对特征进行处理后输出分类的标签的概率最后得到标签。将经过线性注意力处理的特征通过线性层处理,将高维特征映射到类别维度,输出分类概率。

2.3 词频隐因子模块

对于文本分类任务,文本特征表示所包含语义信息的质量决定着任务效果的好坏。词频隐因子模块的作用是使用矩阵分解的方法获取文本的词频隐因子,和预训练模型的特征共同构成文本特征。此模块主要包含两个步骤构建词频矩阵和对词频矩阵进行分解获取词频隐因子。

词频矩阵是一个表示文本数据中各个单词在不同标签下出现频率的矩阵,其中行对应单词,列对应标签,每个元素表示某个单词在对应标签中的词频。词频矩阵的构建方法如下,设训练数据集 D_{train} 包含 m 个文本样本,每个文本样本 t_m 有其对应的标签 c_k。将所有文本样本组成集合 $\boldsymbol{T}=[t_1, t_2, \cdots, t_m]$,

标签集合为$\boldsymbol{L}=[l_1,l_2,\cdots,l_k]$。然后,从所有文本样本中提取所有不重复单词,构建词汇表$\boldsymbol{W}=[w_1,w_2,\cdots,w_n]$,其中$n$表示所有不重复单词的总数。同时,提取训练集中所有不同的标签,构成集合$\boldsymbol{C}=\{c_1,c_2,\cdots,c_p\}$,其中$p$为标签的总数。

在此基础上,统计每个单词在每个标签中出现的频率。对于每个训练样本t_i及其对应标签l_i,将每个单词的出现次数累加到相应标签的计数中。统计完成后,构建一个词频矩阵$\boldsymbol{V}$,其行索引为单词w,列索引为标签c。对于每个单词$w\in\boldsymbol{W}$和标签$c\in\boldsymbol{C}$,计算该单词在该标签中的频率。设$n(w,c)$为单词w在标签c中出现的次数,$n(w)$为单词w在所有标签中出现的总次数,则其在标签c中的频率可以表示为

$$\mathrm{Freq}(w,c)=\frac{n(w,c)}{n(w)} \tag{1}$$

当$n(w)=0$时,表明该单词未出现在训练数据中。通过这种方式,可以得到一个归一化的词频矩阵,便于后续分析,词频矩阵的构建如算法1所示。

算法1　构建词频矩阵

输入:文本内容和标签

输出:词频矩阵$\boldsymbol{V}$

(1) // 步骤1统计所有不重复的单词和标签

(2) for$(t_i,\boldsymbol{C})$ in D_{train} do

(3) 将t_i分割为单词列表$\boldsymbol{W}$ // 计算单词在当前文本中的出现次数

(4) for w in $\boldsymbol{W}$ do

(5) world_label_counts$[w][\boldsymbol{C}]+=1$

(6) end for // 不换行的单行注释

(7) end for

(8) // 步骤2填充词频

(9) for w in unique_words do

(10) total_count$=\sum$world_label_counts$[w]$.values( ) //计算该单词在所有标签中的总出现次数

(11) if total_count>0 then

(12) for(word, count) in

(13) $\boldsymbol{V}[w,\boldsymbol{C}]=\dfrac{\text{world\_label\_counts}[w][\boldsymbol{C}]}{\text{total\_count}}$ //计算词频

(14) end for

(15) end for

(16) end for

(17) return 词频矩阵$\boldsymbol{V}$

然后使用非负矩阵分解对词频矩阵$\boldsymbol{V}$进行分解。设矩阵$\boldsymbol{V}$的维度为$n\times p$,其中n为单词的数量,p为标签的数量。非负矩阵分解(Non-negative matrix factorization, NMF)的目标是找到两个非负矩阵$\boldsymbol{W}$和$\boldsymbol{H}$,使得$\boldsymbol{V}$近似等于$\boldsymbol{W}$和$\boldsymbol{H}$的乘积,有

$$\boldsymbol{V}_{n\cdot p}\approx\boldsymbol{H}_{n\cdot z}\cdot\boldsymbol{W}_{z\cdot p} \tag{2}$$

式中z为分解的秩。

在分解过程中,通过最小化$\boldsymbol{V}$与$\boldsymbol{W}\cdot\boldsymbol{H}$之间的差异来学习词频和标签之间的关系,使用Frobenius

范数作为优化目标，有

$$\min_{W,H} \| V - W \times H \|_{\mathrm{F}}^{2} \tag{3}$$

式中$\| \cdot \|_{\mathrm{F}}$表示Frobenius范数。Frobenius范数是矩阵分析中的一种常用矩阵范数，它用于衡量V和$W \cdot H$两个矩阵之间的差异，确保重构矩阵$W \cdot H$与原始矩阵V尽可能接近，从而实现对数据的有效表示[27]。

通过这种矩阵分解，得到两个矩阵：单词-成分矩阵W和成分-标签矩阵H。矩阵W的每个元素W_{np}反映了单词n在各个隐主题中p的相关性，如果W_{n1}的值比较大则表明单词W_n与隐主题1的相关性较强。而矩阵H的每个元素H_{pz}则反映了隐主题p在标签z的重要性[28]。具体来说，W反映了各单词在不同成分上的权重分布，而H则展示了各成分在不同标签中的重要性。

然后根据词频对文本中的单词进行从高到低排序。对于数据集D_{train}中的文本Text，将其拆分为单词列表，计算每个单词的出现频率。随后，对所有单词按照频率从高到低进行排序，取前k个单词记为W_{Top_k}，后s个单词记为W_{Bottom_s}，并确保这些单词在词频矩阵W中存在。如果提取的高频单词不足k或s个，则用Space进行填充。

最后，将这些单词转化为特征向量。对于每个文本中的单词，提取其在矩阵W中对应的成分向量，如果是使用SPC填充的的单词则特征向量补充为0，然后将这些单词的向量连接成一个完整的特征向量。对于每个文本，特征向量的长度由k加s个单词的成分列数决定。通过这种方式，每个文本生成了一个固定长度的词频隐因子向量。

通过上述处理，本文能够获得文本的矩阵分解特征向量表示。此方法通过提取文本中最具代表性的高频单词，并利用其潜在成分表示，有效降低了文本数据的维度，同时提升了特征的可解释性和分类的效果。

2.4 预训练特征模块

预训练特征模块的核心是使用BERT预训练模型获取文本特征，并将词频隐因子与预训练文本特征融合。

BERT文本特征的提取如图2所示[29]。输入的文本text首先经过嵌入的方式生成Token嵌入并与Segment嵌入和Position嵌入求和得到BERT预训练模型的输入E_N[30]，计算方法为

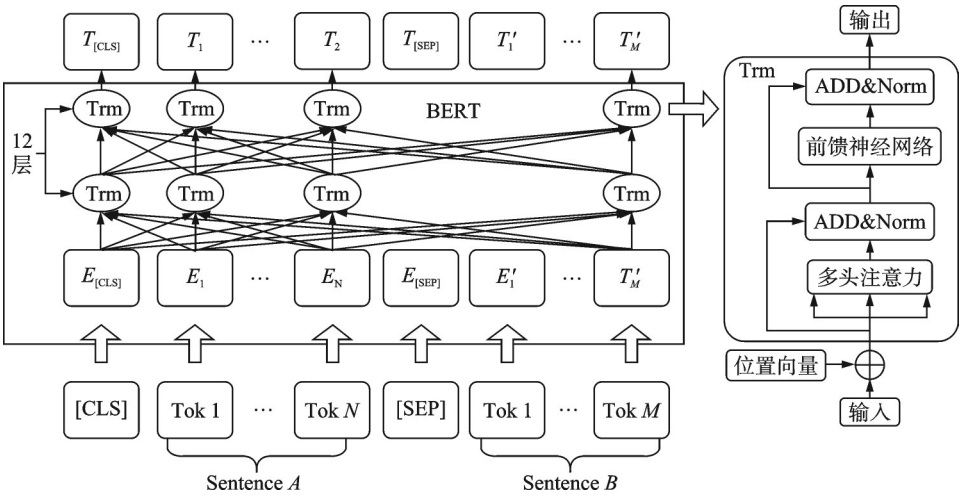

图2 BERT特征提取[29]

Fig.2 BERT feature extraction[29]

$$E(N)=E_{\text{tok}}(N)+E_{\text{seg}}(N)+E_{\text{pos}}(N) \tag{4}$$

然后BERT预训练模型中首个Transformer层会根据初始输入$E(N)$生成3个权重矩阵[31]:查询矩阵$\boldsymbol{W}_{\text{query}}$、键矩阵$\boldsymbol{W}_{\text{key}}$、和值矩阵$\boldsymbol{W}_{\text{value}}$,并利用这3个矩阵分别于输入$E(N)$相乘得到队形的向量$\boldsymbol{Q}_{(0)}$、$\boldsymbol{K}_{(0)}$和$\boldsymbol{V}_{(0)}$,有

$$\boldsymbol{Q}_{(0)}=E(N)\boldsymbol{W}_{\text{query}}\ ,\ \boldsymbol{K}_{(0)}=E(N)\boldsymbol{W}_{\text{key}}\ ,\ \boldsymbol{V}_{(0)}=E(N)\boldsymbol{W}_{\text{value}} \tag{5}$$

再经过注意力机制计算当前Transformer层的输出$T_{\text{output}}^{(0)}$,并将其作为下一层Transformer的输入$T_{\text{inpt}}^{(1)}$以参与新一轮的计算,其过程为

$$T_{\text{input}}^{(1)}=T_{\text{output}}^{(0)}=\text{Softmax}\left(\frac{\left(\boldsymbol{Q}_{(0)}\boldsymbol{K}_{(0)}\right)^{\text{T}}}{\sqrt{d_k}}\boldsymbol{V}_{(0)}\right) \tag{6}$$

$$\boldsymbol{Q}_{(1)}=T_{\text{input}}^{(1)}\boldsymbol{W}_{\text{query}}\ ,\ \boldsymbol{K}_{(1)}=T_{\text{input}}^{(1)}\boldsymbol{W}_{\text{key}}\ ,\ \boldsymbol{V}_{(1)}=T_{\text{input}}^{(1)}\boldsymbol{W}_{\text{value}} \tag{7}$$

式中:$\boldsymbol{W}_{\text{query}}$、$\boldsymbol{W}_{\text{key}}$和$\boldsymbol{W}_{\text{value}}$均为可训练参数矩阵;$d_k$为矩阵$W_{\text{key}}$的维度,其作用在于防止点积结果过大,进而导致梯度过小的问题。得益于BERT中的12层Transformer结构,输入中的每个信息单元均与其他信息单元充分地进行了信息交互[29]。具体而言,文本间的浅层关系通过Transformer结构中的多头注意力机制进行计算,从而生成交互后的丰富特征表示[32]。BERT生成的特征表示作为3个并行模块的共享特征,为各模块共同使用与优化提供支持。

输入文本text经过上述处理之后会生成文本的预训练特征$\boldsymbol{H}_{\text{BERT}}$,接下来与词频隐因子$\boldsymbol{H}_{iw}$进行拼接融合处理得到包含了词频隐因子和预训练文本特征的文本特征$\boldsymbol{H}_V$,其计算方法为

$$\boldsymbol{H}_V=\boldsymbol{H}_{iw}+\boldsymbol{H}_{\text{BERT}} \tag{8}$$

通过上述方法,使用BERT预训练模型获取了文本的预训练特征,并使用拼接的融合方法将词频隐因子和预训练模型特征进行融合得到更丰富的文本特征,为后续的处理提供基础。

2.5　线性注意力模块

线性注意力是一种相对较新的注意力机制,用来提高传统自注意力机制的计算效率,尤其是在处理长序列数据时。传统的自注意力在计算时需要对输入序列中的每一对元素进行交互,计算复杂度为$O(n^2)$,其中n为输入序列的长度。当序列长度变长时,计算成本会大幅增加,导致在处理长文本或序列时效率较低[33]。本文使用线性注意力机制对文本的特征进行处理,获取文本关键特征的同时提升计算效率。

由于模型在BERT预训练的文本特征基础上,增加了词频隐因子的文本特征,文本序列的长度随之增加,进一步提升了计算成本。为解决这一问题,线性注意力机制提供了一种有效方案。通过引入线性投影矩阵,线性注意力机制能够将计算复杂度从$O(n^2)$降低至$O(n)$,显著提升计算效率。其核心思想是将键和值矩阵通过线性投影压缩到较低维度,从而减少计算量,它们的对比如图3所示。

在标准注意力机制中,每个输入序列中的位置都会生成3个向量:查询$\boldsymbol{Q}$、键$\boldsymbol{K}$和值$\boldsymbol{V}$。这些向量用于计算不同位置间的相似度即注意力权重[34]。注意力权重由查询与键向量的点积决定,可表示为

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\text{T}}}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{9}$$

式中:d_k为键向量的维度。该公式的复杂度主要来自$\boldsymbol{Q}\boldsymbol{K}^{\text{T}}$的计算,这要求对每个输入位置与序列中的所有位置进行成对计算,因此复杂度为$O(n^2)$。

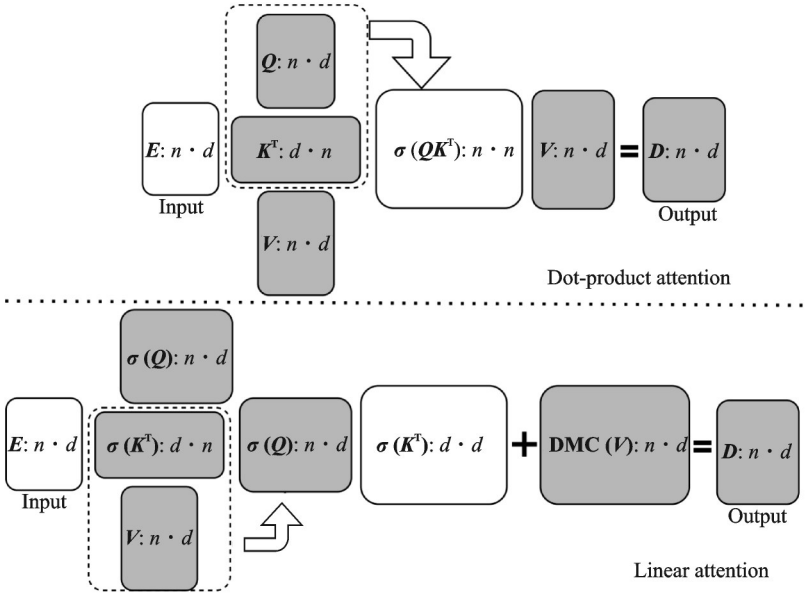

图3 标准注意力和线性注意力对比图

Fig. 3 Contrast diagram of standard and linear attentions

线性注意力将键$\boldsymbol{K}$和值$\boldsymbol{V}$通过线性投影矩阵$\boldsymbol{P}\in\mathrm{R}^{n\times k}$压缩到更低的维度$k$,其中$k\ll n$,从而减少计算量[35]。具体地,键和值的变换公式为

$$\boldsymbol{K}'=\boldsymbol{P}_K\boldsymbol{K} \tag{10}$$

$$\boldsymbol{V}'=\boldsymbol{P}_V\boldsymbol{V} \tag{11}$$

新的注意力机制变为

$$\mathrm{Attention}(\boldsymbol{Q},\boldsymbol{K}',\boldsymbol{V}')=\mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}'^{\mathrm{T}}}{\sqrt{d_k}}\right)\boldsymbol{V}' \tag{12}$$

由于k的值远小于n,线性注意力将注意力的计算复杂度从$O(n^2)$降低到$O(nk)$,极大地提高了处理长序列的效率。

为了进一步减少计算和内存消耗,线性注意力还引入了共享键和值投影矩阵的概念。在某些变体中,键和值共享同一个投影矩阵$\boldsymbol{P}$,可以表示为式(13)。这不仅减少了模型的参数数量,还进一步简化了计算过程,有

$$\boldsymbol{K}'=\boldsymbol{P}\boldsymbol{K}\qquad\boldsymbol{V}'=\boldsymbol{P}\boldsymbol{V} \tag{13}$$

通过对键和值矩阵进行线性化投影,降低了自注意力机制的复杂度,模型可以更高效地处理长序列数据[36]。与标准自注意力机制相比,线性注意力机制保持了特征提取的能力,同时极大地提高了计算效率,是长序列任务中的有效解决方案。

拼接后的文本特征$\boldsymbol{H}_V$通过线性注意力机制进行处理。将原本需要高复杂度计算的注意力机制降为线性复杂度,极大地提升了处理长序列输入的效率。经过多头注意力机制的处理,模型生成了一个新的上下文表示。在生成的上下文向量中,模型提取第一个位置(即[CLS]位置)的嵌入,作为句子的全局特征记为$\boldsymbol{Y}$。

通过上述方法,线性注意力机制显著降低了计算复杂度,使得长序列的注意力计算更加高效。同时,该机制能够保留 BERT 强大的特征提取能力,并合理利用额外特征,提升了模型的整体表现。

2.6　分类模块

分类模块的主要任务是对经线性注意力机制处理后的加权特征处理,从而得到文本对应标签的概率分布。本文使用线性分类器对加权特征进行处理,其是深度学习模型中常见的输出层结构,负责将输入特征映射到目标类别。核心步骤包括:输入特征与权重矩阵的乘积、添加偏置项生成logist,通过softmax函数转换为概率分布,并通过交叉熵损失来优化模型的参数。模型的整体工作机制如算法2所示。

算法2　模型分类

输入:文本和标签。

输出:分类结果。

(1) for 每条文本 t_i in 数据集 D_{train} do // 不换行的单行注释

(2) $H_{iw}=H_{\text{Top\_1}}+\cdots+H_{\text{Top\_}k}+H_{\text{Bottom\_1}}+\cdots+H_{\text{Bottom\_}s}$ // 获取词频矩阵文本特征

(3) $\boldsymbol{H}_{\text{BERT}}=\text{bert\_model}(t_i,\text{att\_mask})$ //获取BERT预训练文本特征

(4) end for

(5) $\boldsymbol{H}_V=\boldsymbol{H}_{iw}+\boldsymbol{H}_{\text{BERT}}$ // 融合文本特征

(6) $Y_M=\text{linformer}(H_V)$ // 通过线性注意力机制处理

(7) for 每条文本的特征 in 数据集 D_{train} do

(8) cls_output = $\boldsymbol{Y}$ //提取文本的表示

(9) **logist** = classifier(cls_output) //通过分类器生成**logist**

(10) 使用交叉熵损失函数计算损失 //计算损失

(11) 反向传播误差并更新模型参数 //反向传播

(12) end for

(13) return 分类结果 logist //输出分类结果

线性分类器的输入是来自线性注意力模块的[CLS]位置特征,该特征表示整个序列的全局表示。将此向量记作 $\boldsymbol{Y}\in\mathbf{R}^{B\times(y_{\text{BERT}}+y_{iw})}$,其中 B 为批次大小,y_{BERT} 为BERT的隐藏层维度,y_{iw} 为外部特征的维度。

线性分类器的核心操作是线性变换,表示为矩阵乘法加上偏置。给定输入特征 $\boldsymbol{Y}$,线性分类器输出的**logist**计算公式为

$$\text{logist}=\boldsymbol{YM}+\boldsymbol{b} \tag{14}$$

式中:$\boldsymbol{M}\in\mathbf{R}^{B\times(y_{\text{BERT}}+y_{iw})\times p}$ 为可训练的权重矩阵;p 为分类任务中的类别数;$\boldsymbol{b}\in\mathbf{R}^{p}$ 为可训练的偏置项[37]。这里是将输入的高维特征向量投影到1个大小为 p 的向量空间,其中每个元素对应1个类别的预测得分。

输入特征向量 $\boldsymbol{Y}$ 通过矩阵乘法与权重矩阵 $\boldsymbol{M}$ 相乘,生成大小为 $[B,p]$ 的向量。其计算方法为

$$\boldsymbol{z}=\boldsymbol{YM} \tag{15}$$

式中:$\boldsymbol{z}\in\mathbf{R}^{B\times p}$ 表示批次中每个样本对于每个类别的得分。

然后为每个类别得分添加偏置项以增加模型的灵活性,最终的**logist**计算公式为

$$\text{logist}=\boldsymbol{z}+\boldsymbol{b} \tag{16}$$

式中:$\boldsymbol{b}$ 表示大小为 p 的偏置项,用于对每个类别的得分进行偏移。具体来说,每个样本的**logist**值为

$$\text{logist}_{i,p}=z_{i,p}+b_p \tag{17}$$

式中:b_p 为与类别 p 相关的偏置项。

在线性分类器生成**logist**后，使用softmax函数将这些**logist**转换为概率分布。softmax函数的定义为

$$d_p=\frac{\exp(\mathbf{logist}_p)}{\sum_{i=1}^{p}\exp(\mathbf{logist}_i)} \tag{18}$$

式中d_p表示样本属于类别p的概率。softmax函数将**logist**转换为正数，并将它们归一化，使得所有类别的概率和为1。模型基于最大概率进行类别预测，有

$$\hat{y}=\operatorname{argmax}(d_p) \tag{19}$$

在训练过程中，模型通过最小化交叉熵损失来更新参数。交叉熵损失衡量模型预测的概率分布与真实标签的差异，其定义为

$$l=-\frac{1}{B}\sum_{i=1}^{B}\sum_{p=1}^{p}y_{i,p}\log(p_{i,p}) \tag{20}$$

式中：B为批次大小；$y_{i,p}$为真实标签的one-hot编码；$p_{i,p}$为模型预测的类别p的概率。通过最小化损失函数，模型能够调整权重矩阵$\boldsymbol{M}$和偏置项$\boldsymbol{b}$，以提升分类性能。

通过这种方式可以将高维特征向量映射到类别空间，利用权重矩阵和偏置项进行线性变换，并通过softmax函数将logits转换为概率分布，进而完成分类任务。通过最小化交叉熵损失，模型得以学习最佳的参数，使其在训练过程中逐步提高分类性能，最终实现对输入文本的准确分类。

3 实验设计

3.1 实验数据

本实验使用了来自AGNews公开数据集的新闻文本、Yahoo！Answer的公开数据集和Stanford Sentiment Treebank 2（SST-2）的公开数据集。AGNews是由康奈尔大学发布的新闻分类数据集，涵盖了4个主要类别：世界新闻、体育新闻、商业新闻和技术新闻。每个类别包含约30 000条新闻。本实验从中随机抽取60 000条作为训练集，4 000条作为测试集，以及4 000条作为预测集。Yahoo！Answer数据集是一个大规模的问答分类数据集，涵盖了10个主要类别：社会与文化、科学、健康、教育、娱乐、体育、商业、计算机与互联网、政治与政府以及其他类别。每个类别包含约140 000条问答文本。本实验从中随机抽取60 000条作为训练集，4 000条作为测试集，以及60 000条作为预测集。SST-2是一个用于情感分析的经典数据集，主要用于对文本进行二分类（正向情感和负向情感）。该数据集包含从电影评论中提取的句子，并已被标注为正向或负向情感标签。本实验从中随机抽取53 681条作为训练集，4 000条作为测试集以及4 000条作为预测集。为确保实验结果不受数据集的不同类别分布的影响，各个类别中的数据量保持一致。表2展示了本次实验所使用的3个数据集。

表2 数据集信息

Table 2 Data set information

数据集	训练集	测试集	预测集
AGNews	60 000	4 000	4 000
Yahoo！Answer	60 000	4 000	4 000
SST-2	53 681	4 000	4 000

3.2 评价指标

本文所有试验使用精确率（Precision）、召回率（Recall）和F_1值作为评价指标[38]对本文模型进行评估。

精确率[39]衡量的是模型预测为正的样本中有多少是实际的正例，反映了模型在正类预测方面的准确度，其计算方法为

$$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}} \tag{21}$$

式中:真阳性(True positive,TP)为模型正确预测为正类的样本数;真阴性(True negative,TN)为模型正确预测为负类的样本数;假阳性(False positive,FP)为模型错误地将负类样本预测为正类;假阴性(False negative,FN)为模型错误地将正类样本预测为。

召回率[40]表示实际为正类的样本中,模型正确预测为正类的样本所占的比例,反映了模型的敏感性或覆盖率,其计算方法为

$$\text{Recall}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{22}$$

F_1值[41](F_1 score)表示精确率和召回率的调和平均值,用来综合评估模型的性能。当精确率和召回率不平衡时,F_1值能够提供更好的衡量,其计算方法为

$$F_1=2\times\frac{\text{Precosion}\times\text{Recall}}{\text{Precosion}+\text{Recall}} \tag{23}$$

3.3　对比算法

为了充分验证提出算法的有效性,本文选择BERT、ERNIE、BERT-CNN和PS-NET这4种文本分类算法作为对比算法。

(1)BERT[10]:一种双向深层变换器模型,通过上下文的双向编码预训练,使得在自然语言处理任务中能够理解句子中的双向信息。

(2)ERNIE[11]:百度提出的一种语言模型,结合了知识图谱进行预训练,增强了模型在处理具有知识背景的文本时的表现。

(3)BERT-CNN[42]:结合BERT进行特征提取与卷积神经网络(Convolutional neural network,CNN)提升分类效果的模型,能够捕捉文本的全局和局部特征。

(4)PS-NET[43]:是一种结合在线知识蒸馏、同行协作和对抗训练的轻量化半监督学习框架,在低资源文本分类和其他文本挖掘任务中显著提升了模型性能。

3.4　实验方法

本实验基于PyTorch深度学习框架,操作系统为Windows11,CPU为14th Gen Intel(R) Core(TM) i7-14700KF @ 3.40 GHz,所用的显卡为NVIDIA GeForce RTX 4070 Super GPU,RAM为32 GB。

对实验结果本文的模型参数设置如下:对于预训练模型采用12层的BERT-Base-Cased作为预训练语言模型,其中向量维度为768,共110×10^6个参数,训练批次大小设置为128;使用Adam优化算法来优化模型,学习率为2e−5。对于词频隐因子取词频前10个和后10个,共20个单词的隐因子作为特征。模型参数设置如表3所示。

表3　模型参数

Table 3　Model parameter

参数	数值
BERT特征维度	768
Dropout值	0.1
学习率	2e−5
优化器	Adam
矩阵分解隐因子	5
高频词k	10
低频词s	10

4　实验结果与分析

4.1　实验结果

表4展示了在3个数据集上不同方法的实验结果,其中加粗的为最佳性能值。从实验结果可以看

出，LTTW算法在所有数据集的所有评估指标（精确率、召回率和F_1分数）上均表现出色，进一步验证了其在文本分类任务上的卓越性能。在AGNews数据集中，LTTW的精确率、召回率和F_1分数分别达到93.73%、93.71%和93.70%，显著高于表现最佳的对比模型PS-NET（92.95%、92.99%和92.93%）。在Yahoo！Answer数据集中，LTTW分别取得了70.91%、70.96%和70.87%的成绩，同样优于表现较好的PS-NET模型（70.78%、70.75%和70.69%）。在SST-2数据集中，LTTW的精确率、召回率和F_1分数分别为93.81%、93.95%和93.82%，全面超越表现最优的对比模型ERNIE（93.01%、92.84%和92.92%）。整体来看，LTTW算法在精确率、召回率和F_1分数上始终保持领先。实验过程中采用了十折交叉验证方法，进一步降低了偶然性并提升了模型的泛化能力。

表4 实验结果

Table 4 Experimental result %

数据集	算法	Precision	Recall	F_1
AGNews	BERT	92.60	92.58	92.58
	ERNIE	92.88	92.83	92.83
	BERT-CNN	91.66	91.63	91.60
	PS-NET	92.95	92.99	92.93
	LTTW	**93.73**	**93.71**	**93.70**
Yahoo! Answer	BERT	70.73	70.68	70.68
	ERNIE	70.28	70.90	70.12
	BERT-CNN	70.23	70.49	70.10
	PS-NET	70.78	70.75	70.69
	LTTW	**70.91**	**70.96**	**70.87**
SST-2	BERT	92.54	92.48	92.51
	ERNIE	93.01	92.84	92.92
	BERT-CNN	91.86	91.12	91.97
	PS-NET	92.86	92.91	92.88
	LTTW	**93.81**	**93.95**	**93.82**

综上所述，LTTW在文本分类任务中的表现优于所有对比模型，尤其在精确率、召回率等关键指标上表现显著，展现了其在文本分类任务中的潜在应用价值。

4.2 消融实验

为了进一步分析模型各个模块的有效性，本文设计了3种模型的变体，并将这些变体的分类性能与完整模型进行比较，实验使用AGNews数据集。实验结果如表5所示。根据实验结果可以发现，尽管各个模块对整体性能的贡献有所不同，但是如果删除其中的任何一个模块都会导致性能下降，这表明这些模块的引入是有意义的。

LTTW-M模型去除了词频隐因子模块，仅使用BERT预训练模型的特征，经线性注意力机制处理后完成文本分类任务。实验数据显示，去除词频隐因子模块后，模型在所有数据集上的性能均有显著下降，其中AGNews数据集的精确率从93.73%降至92.43%，Yahoo！Answer数据集从70.91%降至68.40%，SST-2数据集从94.94%降至93.25%。尽管时间成本略有减少，但性能显著下降。这表明词频隐因子模块在提升模型性能上具有重要作用。该模块通过捕捉文本中高频和低频词汇特征，增强了模型的整体文本理解能力。当该模块被移除后，模型对隐含统计特征的敏感性减弱，导致分类性能下降。

表5　消融实验结果

Table 5　Ablation experiment results

数据集	模型	Precision/%	Recall/%	F_1/%	Time/epoch/s
AGNews	LTTW-M	92.43	92.41	92.39	**106.45**
	LTTW-L	92.84	92.81	92.82	108.43
	LTTW-Att	93.27	93.13	93.15	111.63
	LTTW	**93.73**	**92.71**	**92.70**	109.87
Yahoo! Answer	LTTW-M	68.40	68.31	68.37	**124.60**
	LTTW-L	70.18	70.05	70.12	127.18
	LTTW-Att	70.83	70.80	70.69	132.55
	LTTW	**70.91**	**70.96**	**70.87**	129.37
SST-2	LTTW-M	93.25	93.15	93.18	**86.25**
	LTTW-L	93.61	93.51	93.57	87.51
	LTTW-Att	94.61	94.86	94.72	88.32
	LTTW	**94.94**	**94.84**	**94.89**	86.77

LTTW-L模型去除了线性注意力机制模块,在融合BERT预训练模型特征与词频隐因子特征后,直接进行文本分类任务。实验结果表明,去除线性注意力机制模块后,模型的性能有所下降。AGNews数据集的精确率从93.73%降至92.84%,SST-2数据集从94.94%降至93.61%。时间成本略微增加。这说明线性注意力机制能够有效捕捉全局依赖关系,尤其是在长文本或复杂句子中,远距离词汇之间的语义关联对于分类任务至关重要。线性注意力机制在保证较低计算复杂度的前提下,显著提升了模型对长距离依赖关系的建模能力。

LTTW-Att模型将线性注意力机制替换为传统的多头自注意力机制,结合BERT预训练模型特征与词频隐因子特征,经多头自注意力机制处理后进行文本分类任务。实验结果显示,用传统多头自注意力机制替换线性注意力机制后,模型性能轻微下降,Yahoo! Answer数据集的准确率从70.91%降至70.83%,SST-2数据集从94.94%降至94.61%。时间成本有所增加,在AGNews数据集上从109.87 s增至111.63 s,这表明尽管多头自注意力机制在捕捉全局依赖关系方面表现良好,但计算复杂度较高且时间消耗较大,而线性注意力机制能够以更高的效率捕捉长距离依赖关系,在性能和效率之间达到了更优的平衡。

综上所述,消融实验结果验证了LTTW模型中各模块的重要性与贡献。词频隐因子模块通过补充文本语义特征提升了模型性能,线性注意力机制在降低计算复杂度的同时有效捕捉了全局依赖关系,从而使模型在分类任务中表现更加优异。

4.3　超参数调节

在确保实验数据集和实验环境一致的前提下,本文系统地探讨了不同超参数设置对模型性能的影响,实验使用AGNews数据集。

针对不同数量的词频隐因子对文本特征的作用,分别选取词频隐因子的数量为30、25、20、15和10进行实验。实验结果如表6所示,不同数量的词频隐因子对模型性能有显著影响。当隐因子数量为20时,模型在所有评价指标中精确率93.65%、召回率93.62%和F_1值93.64%均达到最高,说明适量的隐因子能够有效平衡特征捕捉与冗余控制,提升模型性能。隐因子数量过多会引入冗余信息,导致模型泛化能力下降,而隐因子数量过少则难以充分捕捉文本中的语义特征,显著削弱分类性能。因此,隐因

表6 不同词频隐因子数量结果

Table 6 The number results of hidden factors with different word frequency

评价指标	30	25	20	15	10
Precision/%	93.46	93.58	**93.65**	93.36	93.05
Recall/%	93.44	93.52	**93.62**	93.34	93.02
F_1/%	93.41	93.53	**93.64**	93.33	93.303

子数量的合理选择对于模型的优化至关重要,其中20为最优设置。

针对高词频和低词频隐因子对文本特征的作用,分别选取词频隐因子的数量为高频15加低频15、高频10加低频10、高频15加低频5、高频5加低频15和高频5加低频5进行实验。实验结果如表7所示,高词频和低词频隐因子的数量配置对模型性能具有显著影响。当隐因子的高频和低频配置均为10时,模型在各项指标上均表现最佳,具体为精确率93.73%、召回率93.73%、F_1值93.70%。这表明,在高频和低频隐因子之间实现均衡配置能够最大化捕捉文本中关键的语义信息,同时避免因过多高频或低频隐因子导致的冗余或信息缺失。当高频或低频隐因子的数量过多或过少时,模型性能均有所下降,进一步说明均衡配置的重要性。因此隐因子数量的合理分配是提升模型性能的关键,高10低10的配置为最优方案。

表7 不同数量的高频和低频隐因子

Table 7 Different number of high and low frequency hidden factors

评价指标	高15低15	高15低5	高10低10	高5低15	高5低5
Precision/%	93.08	93.60	**93.73**	93.38	93.36
Recall/%	93.06	93.58	**93.73**	93.36	93.34
F_1/%	93.03	93.58	**93.70**	93.37	93.33

为了充分验证线性注意力机制的性能和稳定性,本文对模型中的关键超参数进行了系统性的调节和分析。这些超参数在任务中的表现直接影响模型的收敛速度、计算效率以及最终的分类效果。以下是针对核心超参数设置及其实验结果的详细分析:

(1)模型深度(depth)。线性注意力模块堆叠的层数,对模型的特征提取能力起着至关重要的作用。在实验中,本文分别测试了浅层模型与深层模型的表现,以研究堆叠更多注意力模块是否有助于捕获更复杂的语义信息。实验结果表明,随着模型深度的增加,模型的精确率、召回率和F_1得分均呈现下降趋势。这表明,尽管增加堆叠层数可能增强模型的表征能力,但过深的网络会引发过拟合或导致训练不稳定,从而导致性能的退化。在实验中,depth=1的浅层模型表现最佳,这说明对于当前任务,简单的模型结构已足以提取有效的语义信息。

(2)低秩近似维度(K)。线性注意力的低秩近似维度是模型效率提升的关键参数,决定了原始高维特征空间的降维程度。在实验中,本文分别测试了K=8、16、32、64和128的设置,以评估降维后的信息保持能力与计算效率之间的权衡。实验结果显示,K对模型性能具有显著影响。当K=32时,模型的精确率、召回率和F_1得分均达到最高值,表明适当的降维可以有效平衡信息保持能力和计算效率。然而,当K值过小或过大时,模型性能均有所下降。这一结果表明,过度降维会导致信息丢失,而降维不足则可能增加计算复杂度,降低模型效率。

(3)批量大小(batch_size)。批量大小是影响模型训练效率和稳定性的关键超参数。在实验中,选取了batch_size=16、32、64、128和256的设置,分析了不同批量大小对模型收敛性和泛化性能的影响。结果表

明,批量大小对模型的收敛稳定性和性能具有一定影响。当批量大小为128时,模型性能最佳,说明中等规模的批量大小在收敛效率与性能之间达到了良好的平衡。而较小的批量大小可能导致梯度更新不稳定,影响模型的收敛;较大的批量大小则可能限制模型对训练数据多样性的适应能力。

(4)学习率(lr)。学习率是影响模型收敛速度和最终性能的核心超参数之一。本文基于Adam优化器,将初始学习率设置为多个值进行测试,以确保模型能够在有限时间内高效收敛,同时避免梯度爆炸或梯度消失的问题。实验结果表明,不同学习率对模型性能具有显著影响。当学习率设置为2×10^{-5}时,模型性能达到最佳,表现为最高的精确率、召回率和F_1得分。学习率过小会导致模型收敛速度变慢,而学习率过大会引发梯度更新过快,导致模型性能不稳定。

从实验结果可以看出,每个超参数的选择都需要根据模型的性能进行权衡。较浅的模型深度(depth=1)、中等的降维程度(K=32)、适中的批量大小(batch_size=128)和优化的学习率(lr=2×10^{-5})能够提供最佳的性能表现。通过上述超参数调节实验,本文系统性地分析了这些关键参数对模型性能的影响,从而为线性注意力机制在文本分类任务中的应用提供了全面的优化指导。

表8　线性注意力参数

Table 8　Linear attention parameter

超参数	数值	Precision/%	Recall/%	F_1/%
depth	1	**93.73**	**93.71**	**93.70**
	2	93.47	93.44	93.43
	3	92.82	92.81	92.78
	4	93.32	93.22	93.24
	5	92.80	92.76	92.76
K	8	92.80	92.76	92.76
	16	93.22	93.17	3.17
	32	**93.73**	**93.71**	**93.70**
	64	93.00	92.98	92.96
	128	93.12	93.10	93.09
batch_size	16	93.02	92.92	92.96
	32	93.13	93.15	93.14
	64	93.69	93.62	93.62
	128	**93.73**	**93.71**	**93.70**
	256	93.21	93.14	93.16
lr	1×10^{-5}	93.46	93.42	93.43
	2×10^{-5}	**93.73**	**93.71**	**93.70**
	3×10^{-5}	93.39	93.32	93.34
	4×10^{-5}	93.01	93.01	93.00
	5×10^{-5}	92.88	92.86	92.85

4.4　实验分析

LTTW模型的核心在于通过矩阵分解技术提取文本的词频隐因子,并将其与预训练的BERT模型相融合,从而有效提高文本分类任务的性能。本文提出的LTTW方法首先使用矩阵分解技术提取文本中前k个高频词和前s个低频词的词频隐因子,这一过程捕捉了文本中的显著词汇,从而提高了模型的特征表示能力。相比之下,传统基于词嵌入的模型如BERT、ERNIE和BERT-CNN,主要依赖于全局词嵌入或卷积神经网络捕捉上下文信息,虽然能够处理复杂的句子结构,但对词频等直接影响文本分类的关键信息提取不如LTTW精确。通过将提取的词频隐因子与预训练模型的特征相融合,LTTW不仅能够有效利用BERT在上下文理解和语义层面的优势,还能够综合词频隐因子,保证了对文本全局语义和局部词汇信息的全面捕捉。这一融合策略在分类任务中表现出了明显优势,表现在实验结果中的4项评估指标均优于其他模型,特别是与具有代表性的BERT模型相比,LTTW的精确率有明显提升。

LTTW进一步采用线性注意力机制来处理融合后的特征。线性注意力机制相比传统的全局注意力机制,计算复杂度更低,能够更高效地处理长文本序列。线性注意力的优势在于其简化了上下文信息的捕捉,使得特征权重分配更加直接,并显著降低了计算成本,同时保持了模型的性能。这种改进可以使LTTW在性能提升的同时,仍然保持较高的计算效率,特别是在大规模数据集上的应用中,线性注意力机制使得LTTW在保证分类准确性的同时,不会带来计算资源的过度消耗。实验结果中LTTW的精确率和F_1分数均高于ERNIE,表明线性注意力在处理融合特征时,能够在性能上实现更优的平衡。

本文提出的LTTW模型尽管其在文本分类任务中表现出了显著的优势,但也存在一些局限性。LTTW模型通过矩阵分解技术提取词频隐因子,这种方法依赖于文本的词频分布来捕捉关键信息。然而,词频隐因子在特定领域内可能表现出较好的效果,但在跨领域任务中,词频特征的泛化能力可能受到限制。在不同领域或不同语言的文本中,词频的分布特性可能存在较大差异,因此LTTW在处理跨领域文本时,可能需要针对不同的领域进行模型调整或重新训练,以确保模型能够有效适应不同的词汇分布。

总体来看,本文提出的LTTW模型通过矩阵分解提取词频隐因子,并将其与BERT预训练模型融合,增强了文本内容特征的丰富度。采用线性注意力机制对特征进行高效处理,有效捕捉文本关键特征的同时提升了序列处理的效率。实验结果表明,该模型能够显著提升文本分类的性能,具有较大的实际应用价值。

5 结束语

本文提出了一种融合词频隐因子与文本特征的线性注意力分类模型,通过矩阵分解提取词频隐因子,并将其与BERT预训练模型融合,增强了文本内容特征的丰富度,使用线性注意力机制对特征进行高效处理,有效捕捉文本关键特征的同时提升了序列处理的效率,提高了文本分类的精确率和效率。

LTTW模型的设计特点使其在跨语言文本分类任务中具有一定的应用潜力。通过词频隐因子提取潜在语义信息的能力,可以在多语言环境中捕捉语言间的共有语义特征,但是需要根据不同语言的特点对文本进行处理;同时,引入多语言预训练模型替代单语言模型,有助于提升对不同语言的泛化能力。此外,线性注意力机制在捕捉全局依赖关系的同时具有高效性,可支持长序列文本的跨语言处理。未来的研究可以进一步探讨基于多语言词汇对齐方法,如共享嵌入空间或翻译对齐优化词频隐因子表达的效果,并通过跨语言数据集测试验证模型在低资源语言和跨语言迁移任务中的性能,这将为模型在全球化场景中的应用,如多语言评论分类和社交媒体分析提供更广泛的支持。

未来随着文本分类任务应用领域的扩大,对文本分类的准确率和效率的需求将持续增长。本文提出的模型在未来可以进一步探索以下几个方向。首先,当前模型对词频隐因子和BERT预训练特征的融合方式较为简单,通过更加复杂的融合网络或自适应权重分配机制,可以进一步优化不同特征的协同作用,使模型在处理不同类型文本时具有更强的鲁棒性。其次,尽管本模型使用了线性注意力机制来降低计算复杂度,但在处理超长文本时,仍然可能遇到效率瓶颈。未来的研究可以探讨更轻量化的预训练模型,或优化词频隐因子提取方法,进一步降低计算资源的需求,从而实现更加高效的文本分类模型。最后,随着文本分类应用的扩展,用户对于模型可解释性的需求越来越高,未来可以结合可解释性分析工具或开发新的可解释性机制,帮助用户理解模型的决策过程,从而提高其在实际应用中的信任度和可用性。

参考文献:

[1] HUA Y C, DENNY P, WICKER J, et al. A systematic review of aspectbased sentiment analysis: Domains, methods, and trends[J]. Artifcial Intelligence Review, 2024, 57(11): 1-49.

[2] TAN K L, LEE C P, LIM K M. A survey of sentiment analysis: Approaches, datasets, and future research[J]. Applied Sciences-Basel, 2023,13(7): 1-21.

[3] VENKIT P N, SRINATH M, GAUTAM S, et al. The sentiment problem: A critical survey towards deconstructing sentiment analysis[EB/OL]. (2023-10-18)[2024-11-15]. https://arxiv.org/abs/2310.12318.

[4] SALAKHUTDINOV R, MNIH A. Probabilistic matrix factorization[C]//Proceedings of the 20th International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc.,2007: 1257-1264.

[5] ESFAHANI S H N, ADDA M. Classical machine learning and large models for text-based emotion recognition[J]. Procedia Computer Science, 2024,241: 77-84.

[6] WU Z, F. LIU N, POTTS C. Identifying the limits of cross-domain knowledge transfer for pretrained models[C]//Proceedings of the 7th Workshop on Representation Learning for NLP. Dublin, Ireland: Association for Computational Linguistics, 2022: 100-110.

[7] SUTSKEVER I, MARTENS J, HINTON G. Generating text with recurrent neural networks[C]//Proceedings of the 28th International Conference on International Conference on Machine Learning. Madison, WI, USA: Omnipress, 2011: 1017-1024.
[8] DENG S, LI Q, DAI R, et al. A chinese power text classifcation algorithm based on deep active learning[J]. Applied Soft Computing, 2024, 150:111067.
[9] TURTON J, SMITH R E, VINSON D. Deriving contextualised semanticfeatures from BERT (and other transformer model) embeddings[C]//Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021). [S.l.]: Association for Computational Linguistics, 2021: 248-262.
[10] DEVLIN J, CHANG M W, LEE K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding [C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, 2019: 4171-4186.
[11] ZHANG Z, HAN X, LIU Z, et al. ERNIE: Enhanced language representation with informative entities[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, 2019: 1441-1451.
[12] BROWN T B, MANN B, RYDER N, et al. Language models are few-shot learners[C]//Proceedings of the 34th International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc., 2020: 25.
[13] JIAO X, YIN Y, SHANG L, et al. TinyBERT: Distilling BERT for natural language understanding[C]//Proceedings of Findings of the Association for Computational Linguistics: EMNLP 2020. [S.l.]: Association for Computational Linguistics, 2020: 4163-4174.
[14] SHENG P, SHI Y, LIU X, et al. LSNet: Real-time attention semantic segmentation network with linear complexity[J]. Neurocomputing, 2022, 509: 94-101.
[15] GRAVES A. Long short-term memory[M]. Berlin, Heidelberg: Springer, 2012: 37-45.
[16] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc., 2017: 6000-6010.
[17] LIU T, HU Y, GAO J, et al. Multi-modal long document classifcation based on hierarchical prompt and multi-modal transformer[J]. Neural Networks, 2024, 176: 106322.
[18] ROGERS A, KOVALEVA O, RUMSHISKY A. A primer in BERTology: What we know about how BERT works[J]. Transactions of the Association for Computational Linguistics, 2021, 8: 842-866.
[19] JIANG Z, YANG M, TSIRLIN M, et al. "low-resource" text classifcation: A parameter-free classifcation method with compressors[C]//Proceedings of Findings of the Association for Computational Linguistics: ACL 2023. Toronto, Canada: Association for Computational Linguistics, 2023: 6810-6828.
[20] 杨卓峰,李旸,李德玉.一种半监督金融事件多标签分类方法[J].数据采集与处理,2024,39(2): 385-394.
YANG Zhuofeng, LI Yang, LI Deyu. A semi-supervised multi-label classification method for financial events[J]. Journal of Data Acquisition and Processing, 2024, 39(2): 385-394.
[21] CESARINI M, MALANDRI L, PALLUCCHINI F, et al. Explainable ai for text classifcation: Lessons from a comprehensive evaluation of post hoc methods[J]. Cognitive Computation, 2024, 16(6): 1-19.
[22] MUINONEN K, PENTTILÄ A. Scattering matrices of particle ensembles analytically decomposed into pure mueller matrices [J]. Journal of Quantitative Spectroscopy and Radiative Transfer, 2024, 324: 109058.
[23] HE Z, LIN Y, LIN Z, et al. Multi-label feature selection via similarity constraints with non-negative matrix factorization[J]. Knowledge-Based Systems, 2024, 297: 111948.
[24] 陈玮,卢佳伟.基于特征矩阵优化与数据降维的文本聚类算法[J].数据采集与处理,2021,36(3): 587-594.
CHEN Wei, LU Jiawei. Text clustering algorithm based on feature matrix optimization and data dimensionality reduction[J]. Journal of Data Acquisition and Processing, 2021, 36(3): 587-594.
[25] MAO C, WU Z, LIU Y, et al. Matrix factorization recommendation algorithm based on attention interaction[J]. Symmetry, 2024, 16(3): 267.
[26] DHAL P, AZAD C. A fine-tuning deep learning with multi-objective-based feature selection approach for the classifcation of text[J]. Neural Computing and Applications, 2024, 36(7): 3525-3553.
[27] LIU Z, HU D, WANG Z, et al. LatLRR for subspace clustering via reweighted frobenius norm minimization[J]. Expert Systems with Applications, 2023, 224: 119977.
[28] LI X J, DENG G S, WANG X Z, et al. A hybrid recommendation algorithm based on user comment sentiment and matrix decomposition[J]. Information Systems, 2023, 117: 102244.

[29] 李芳芳, 苏朴真, 段俊文, 等. 多粒度信息关系增强的多标签文本分类[J]. 软件学报, 2023, 34(12): 5686-5703.
LI Fangfang, SU Puzhen, DUAN Junwen, et al. Multi-label text classification with enhancing multi-granularity information relations[J]. Journal of Software, 2023, 34(12): 5686-5703.
[30] ZHOU X, HUANG H, CHI Z, et al. RS-BERT: Pre-training radical enhanced sense embedding for Chinese word sense disambiguation[J]. Information Processing Management, 2024, 61(4): 103740.
[31] KWON N, YOO Y, LEE B. Class conditioned text generation with style attention mechanism for embracing diversity[J]. Applied Soft Computing, 2024, 163: 111893.
[32] RODRAWANGPAI B, DAUNGJAIBOON W. Improving text classification with transformers and layer normalization[J]. Machine Learning with Applications, 2022, 10: 100403.
[33] ZHANG F, GUO T, WANG H. DFNet: Decomposition fusion model for long sequence time-series forecasting[J]. Knowledge-Based Systems, 2023, 277: 110794.
[34] BAI Y, WANG H, HE J. Blin: A multi-task sequence recommendation based on bidirectional KL-divergence and linear attention[J]. Mathematics, 2024, 12(15): 2391.
[35] YANG X, TIAN X, WU J, et al. LLAFN-generator: Learnable linear-attention with fast-normalization for large-scale image captioning[J]. Computer Vision and Image Understanding, 2024, 248: 104088.
[36] YE T, CHEN H, REN H, et al. LPT-Net: A line-pad transformer network for efficiency coal gangue segmentation with linear multi-head self-attention 24mechanism[J]. Measurement, 2024, 226: 114043.
[37] ZHU B, PAN W. Chinese text classifcation method based on sentence information enhancement and feature fusion[J]. Heliyon, 2024, 10(17): e36861.
[38] DUBE L, VERSTER T. Enhancing classifcation performance in imbalanced datasets: A comparative analysis of machine learning models[J]. Data Science in Finance and Economics, 2023, 3: 354-379.
[39] YACOUBY R, AXMAN D. Probabilistic extension of precision, recall, and F_1 score for more thorough evaluation of classifcation models[C]//Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems. [S. l.]: Association for Computational Linguistics, 2020: 79-91.
[40] NOULAPEU NGAFFO A, CHOUKAIR Z. A deep neural network-based collaborative fltering using a matrix factorization with a twofold regularization[J]. Neural Computing and Applications, 2022, 34(9): 6991-7003.
[41] LI F, HU Y. Improving the accuracy of deep learning modelling based on statistical calculation of mathematical equations[C]//Proceedings of Advanced Intelligent Computing Technology and Applications. Singapore: Springer Nature Singapore, 2023: 343-353.
[42] ZHANG J, HAO K, SONG TANG X, et al. A multi-feature fusion model for Chinese relation extraction with entity sense[J]. Knowledge-Based Systems, 2020, 206: 106348.
[43] MAO Q, JIANG W, LIU J, et al. Lightweight contenders: Navigating semi-supervised text mining through peer collaboration and self transcendence[EB/OL]. (2024-12-01)[2024-11-15]. https://arxiv.org/abs/2412.00883.

作者简介:

苏湛(1983-),女,博士,副教授,研究方向:智能控制、推荐系统、复杂网络,E-mail: suzhan@usst.edu.cn。

张旭(1998-),**通信作者**,男,硕士研究生,研究方向:推荐系统、自然语言处理,E-mail:zx_usst@163.com。

艾均(1980-),男,博士,副教授,研究方向:推荐系统、社会计算、通用人工智能,E-mail:aijun@usst.edu.cn。

徐温果(2003-),男,本科生,研究方向:推荐系统、分布式系统。

(编辑:刘彦东)

Linear Attention Text Classification by Combining Text Features and Word Frequency Implicit Factors

SU Zhan, ZHANG Xu*, AI Jun, XU Wenguo

(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: In text classification tasks, effectively extracting text features while improving computational efficiency is a critical challenge. However, traditional methods often struggle to balance feature richness and computational efficiency. To address this issue, this paper proposes a novel text classification model, i.e., the linear attention text classification by combining text features and word frequency implicit factors (LTTW), which introduces a linear attention mechanism to capture key features in the text. Specifically, the model leverages non-negative matrix factorization (NMF) to extract word frequency implicit factors from the term frequency matrix, capturing latent semantic information. Simultaneously, it utilizes pre-trained models to extract semantic features of the text, which are then fused with the word frequency implicit factors to construct a richer text representation. Based on this representation, the linear attention mechanism is applied to effectively capture global dependencies and enhance the processing efficiency of long text sequences. Experiments conducted on public datasets demonstrate that the proposed model outperforms mainstream methods in terms of both accuracy and computational efficiency, with particularly significant efficiency advantages when handling long sequences. The study highlights that the integration of word frequency implicit factors complements the shortcomings of pre-trained models in semantic feature extraction, while the linear attention mechanism effectively captures key textual features and improves sequence processing efficiency. Together, these contributions significantly enhance the performance and efficiency of text classification.

Highlights:

1. Propose a linear attention text classification model that fuses text features with word frequency implicit factors (LTTW). It extracts word frequency implicit factors via non-negative matrix factorization (NMF) and combines them with semantic features from a pre-trained model to construct a richer text representation, thereby compensating for the limitations of using a pre-trained model alone for semantic extraction.

2. Introduce a linear attention mechanism to capture global dependencies in text features. This mechanism improves the processing efficiency of long text sequences while effectively preserving key semantic information.

3. Demonstrate through experiments on public datasets that the proposed model outperforms existing mainstream methods in both classification accuracy and computational efficiency, with a particularly significant efficiency advantage when handling long-sequence data.

Key words: text classification; linear attention mechanism; implicit factors; text features; matrix factorization

Foundation item: National Natural Science Foundation of China (No.61803264).

Received: 2024-12-04; **Revised**: 2025-02-18

***Corresponding author**, **E-mail**: zx.usst@163.com.