2019 年 5 月 第 5 期
Vol. 40 No. 5 2019

基于 Word2vec 和改进型 TF-IDF 的卷积神经网络文本分类模型

王根生[1,2,3],黄学坚[1]

[1](江西财经大学 计算机实践教学中心,南昌 330013)
[2](江西财经大学 国际经贸学院,南昌 330013)
[3](江西财经大学 人文学院,南昌 330013)
E-mail: wgs74@126.com

摘 要: 针对传统机器学习文本分类算法语义特征表达弱、文本表示维度高、词序丢失、矩阵稀疏等问题,提出基于 Word2vec、改进型 TF-IDF 和卷积神经网络三者相结合的文本分类模型(CTMWT):首先通过 Word2vec 模型训练得出样本中所有的词向量;然后提出基于类频方差改进型 TF-IDF 算法,分析每个词向量在文本中的权重,构建基于词向量和权重的文本向量表示;最后借助卷积神经网络从局部到全局相关性特征的学习能力,对该大量文本向量进行深度学习. 试验结果表明三者结合的文本分类模型不仅能实现文本的准确分类,并且相比传统的机器学习文本分类算法具有更好的分类效果.

关 键 词: Word2vec; 改进型 TF-IDF 算法; 卷积神经网络; 文本分类; CTMWT
中图分类号: TP391 **文献标识码**: A **文 章 编 号**: 1000-1220(2019)05-1120-07

Convolution Neural Network Text Classification Model Based on Word2vec and Improved TF-IDF

WANG Gen-sheng[1,2,3], HUANG Xue-jian[1]

[1](Computer Practice Teaching Center, Jiangxi University of Finance and Economics, Nanchang 330013, China)
[2](School of International Trade and Economics, JiangXi University of Finance and Economic, Nanchang 330013, China)
[3](School of Humanities, JiangXi University of Finance and Economic, Nanchang 330013, China)

Abstract: Aiming at the problems of weak semantic feature expression, high dimension of text representation, word order loss and sparse matrix in traditional machine learning text classification algorithm, a text classification model (CTMWT) based on Word2vec, improved TF-IDF and convolution neural network is proposed. Firstly, all the word vectors in the sample are obtained through Word2vec model training. Then an improved TF-IDF algorithm based on class frequency variance is proposed to analyze the weight of each word vector in the text and construct a text vector representation based on word vector and weight. Finally, a large number of text vectors are deeply learnt by means of the learning ability of convolutional neural network from local to global correlation features. Experimental results show that the text classification model can not only achieve accurate text classification, but also has better classification effect than text classification based on traditional machine learning algorithm.

Key words: Word2vec; improved TF-IDF; convolution neural network; text classification; Convolution neural network Text classification Model based on Word2vec and improved TF-IDF

1 引 言

文本分类是自然语言处理(NLP)领域的一个经典问题,在上个世纪 50 年代就有学者进行了相关研究,提出通过先验规则进行分类;到 80 年代,建立了使用专家知识构建文本分类专家系统;随着 90 年代机器学习的发展,形成了基于人工特征工程的浅层分类模型,该模型中特征工程是极为关键的一个步骤,主要包括文本预处理、特征提取、特征表示、文本表示,该模型如图 1 所示.

1.1 国内外研究现状

基于人工特征工程的浅层分类模型是现今主流的文本分类算法,国内外许多学者对该模型的不同阶段展开了相关研究.

1) 文本预处理阶段. 文本预处理主要包括分词和去除停用词,传统的分词算法主要有基于字符串匹配、基于语义和句

图 1 基于人工特征工程的浅层分类模型
Fig. 1 Shallow classification model based on artificial feature engineering

法、基于统计互信息等方法[1];随着深度学习算法的发展,WordEmbedding + Bi - LSTM + CRF 的深度学习文本分词算法凭借无需构建手工特征和出色的性能,成为了目前主流的分词算法[2]. 停用词处理主要针对文本中高频出现的连词、

收稿日期: 2018-10-22 收修改稿日期: 2018-11-08 基金项目: 国家自然科学基金项目(71461012)资助; 国家社会科学基金项目(17BXW059)资助; 江西省高校人文社会科学研究一般项目(TQ1404)资助. 作者简介: 王根生,男,1974 年生,博士,副教授,CCF 会员,研究方向为网络舆情、数据挖掘; 黄学坚,男,1990 年生,硕士,工程师,CCF 会员,研究方向为机器学习.

代词、介词等进行去除,因为这些词不仅增加了文本表示的维度,而且对文本分类毫无作用.

2) 特征提取阶段. 特征提取常用的方法有文件频率(Document Frequency,DF)、信息增益(Information Gain,IG)、卡方检测(chi-square test,CHI)等[3]; DF[4]通过阈值,删除低频词,但有些低频词对类别的判决具有很大贡献; IG[5]通过计算特征词对信息熵的影响进行选择,但也会出对分类贡献大的词漏判; CHI[6]是以 X^2 分布为基础,描述特征词与类别的独立性,缺点是对低频词的区分效果不好. 针对这些传统特征提取算法的不足,不少学者提出了相关的改进算法. 赵倩针对 CHI 和 IG 方法的不足,提出一种基于词频、类间和类内文档频的特征选择方法[7]; 李明江通过引入类词频的概念,构建"文档 + 类 + 词"的立方体模型进行特征选择[8]. 随着近年来 Word2vec 的提出,基于 Word2vec 词向量特征选择成为一个研究热点,陈磊基于 Word2vec 进行文本特征选择,并且发现基于该方法的分类效果好于传统的特征选择方法[9]; Lei Zhu 利用 Word2vec 词向量和 IG 方法进行结合,改进 IG 的不足[10]; Dongwen 利用 Word2vec 词向量计算和情感词的余弦相似度进行情感特征词的选取[11].

3) 特征表示阶段. 传统的表示方法有单词网络(WordNet)、独热编码(one-hot encoding)、词频-逆文本频率(TF-IDF)[12]. WordNet 是基于认知语言学的英文字典,通过对不同词性进行编码,组成一个词汇语义网[13]. WordNet 编码保持了语义相似度,但对相邻同义词的差别不能衡量,且它是主观构建的,维护和添加新词的成本高[14]; one-hot encoding[15]把词编码成一个稀疏向量,这种编码方式简单,但无法表达语义信息; TF-IDF 使用词语对文件的重要度进行特征表示[16],但依然无法表达语义信息. 2013 年,Mikolov 基于神经网络提出了 Word2vec 模型[17],通过对词语上下文和语义关系进行建模,将词语映射到低维实数空间,语义相似的词语在这个空间中也相近,这个特性使得 Word2vec 广泛运用于自然语义处理(NLP)中,如聚类、标注、词性分析等任务[18].

4) 文本表示与分类阶段. 文本表示传统的方法为词袋模型(bag-of-words,BOW)[19],该模型把文档看成是一个无序的词语集合,词语间彼此独立,忽略了词语的上下文关系,并且存在高纬度、高稀疏性的问题; 针对这两个问题空间向量模型(Vector Space Model,VSM)[20]在词袋模型的基础上通过特征选择与计算特征权重进行降维和增加稠密性; 文本的分类器大部分都是基于统计学的机器学习算法,如朴素贝叶斯(Naive Bayes)[21]、KNN[22]、SVM[23]、和神经网络[24]等.

5) 研究深入阶段. 在传统的基于人工特征工程的浅层分类模型中,特征表达语义弱、文本表示维度高、词序丢失、矩阵稀疏等问题是影响分类算法性能的重要因素. 随着研究的不断深入,针对这些问题不少学者提出了相关改进算法. 张谦针对特征维度高、语义弱提出 Word2vec 模型和 SVM 结合的分类算法[25]; 张群针对特征矩阵稀疏问题,设计了基于 Word2vec 与 LAD 主题模型相结合的文本分类算法[26]; 吕淑宝针对传统机器学习文本分类算法准确率低和分布不均的问题,提出基于深度学习的文本分类算法[27].

1.2　小　结

通过以上分析发现,不同阶段的算法都几乎存在一些不足,且大部分改进型算法是基于某一个问题进行的局部改进,针对这一现象,本文提出基于 Word2vec、改进型 TP-IDF 和卷积神经网络三者结合的文本分类模型(CTMWT). 使用 Word2vec 进行词语表示,得到的词向量为低维稠密性实数,并且很好的保留了语义信息,由于 Word2vec 无法表达词汇的重要程度,所以引入 TF-IDF 算法计算每个词向量在文本中的权重; TF-IDF 算法中只考虑特征词在整个语料库中出现的频率,忽略了在不同类别中的分布,针对这个问题本文提出基于类频方差改进型 TF-IDF 算法,运用改进型 TF-IDF 算法构成基于词向量和权重的向量文本表示,这种文本表示很好的保留了词语上下文关系; 最后利用卷积神经网络从局部到全局相关性特征的学习能力,对大量文本向量进行深度学习.

2　相关研究基础

2.1　Word2vec 模型

Word2vec 是基于神经网络从大量文本库中得到语义知识的模型,主要有 CBOW 和 Skip-Gram 两种模型. Skip-Gram 模型通过给定的输入词 w_t 来预测其上下文 $S_{w_t}=(w_{t-k},\cdots,w_{t-1},w_{t+1},\cdots,w_{t+k})$,其中 k 为 w_t 上下文窗口大小,即左右选取词的个数,CBOW 模型则是根据上下文 S_{w_t} 去预测 w_t. Skip-Gram 和 CBOW 训练目标优化函数分别如公式(1)和公式(2)所示:

$$L_{Skip\text{-}Gram}=\sum_{w_t\in C}\sum_{-k\leqslant j\leqslant k,j\neq 0}\log p(w_{t+j}\mid w_t) \tag{1}$$

$$L_{CBOW}=\sum_{w_t\in C}\log p(w_t\mid S_{w_t}) \tag{2}$$

其中 C 为文本库中所有的词语,k 为 w_t 上下文窗口大小. Word2vec 模型的建立并不是为了处理新的预测任务,而是为了得到训练后神经网络中隐藏层的参数矩阵,这些隐藏层参数才是 Word2vec 去学习的词向量.

在对 Word2vec 模型进行训练时,样本为(输入词,输出词),Skip-Gram 模型的输入词和输出词分别为 w_t 和 S_{w_t},CBOW 模型与之相反. 神经网络只能处理数值计算,所以需要对词语进行数值编码,常用的方法是基于训练样本库构建词语表(vocabulary),再根据词语表中的索引对词语进行 one-hot 编码,但会发现神经网络的参数矩阵会非常大. 例如,包含 10000 词语的词汇表,每个词 one-hot 编码为 10000 维的向量,如果想构造 300 维的词向量,隐藏层就至少需要 300 个神经元,输入-隐藏层的参数矩阵将会有 10000* 300 = 300 万个参数,在这么大规模的神经网络中进行梯度下降是很慢的,但由于 one-hot 编码只有一个维度的数值为 1,其余的所有为 0,所以只需选取数值为 1 的输入-隐藏层参数进行更新,为了算法的随机性,再从数值为 0 的连接参数随机选取 5-10 个,这样需要更新的参数就大大减少,降低了训练复杂度. 训练后模型的输出为一个概率分布,使得公式(1)或公式(2)和训练样本间的误差最小,训练得到的每个输入-隐藏层的参数数值就是该输入词的词向量. 这种训练方式,会使得有着相似上下文的词语在词向量空间也非常相近,而有着相似上下文的词语他们的语义也是相近.

2.2　卷积神经网络

卷积神经网络(CNN)是深度学习领域中一个重要的算法,在图像处理方面取得了很好的效果. 一个典型的卷积神经

网络由输入和输出以及多个隐藏层构成,隐藏层由卷积层、池化层和全连接层组成,其中卷积层和池化层配合组成卷积组,逐层学习局部到全局的特征,在最终通过若干个全连接层完成分类工作.

1) 卷积层

卷积层是 CNN 的核心,具有权值共享和局部连接特征.卷积层由一组可学习卷积核(kernels)组成,每个卷积核和上层输入数据的不同局部窗口进行卷积计算,如公式(3)所示:

$$s(t) = (X \cdot W)(t) \tag{3}$$

其中 X 为输入,W 为卷积核,操作符($\cdot$)表示卷积,常见的二维卷积如公式(4)所示:

$$s(i,j) = (X^* W)(i,j) = \sum_m \sum_n x(i-m, j-n)\, w(m,n) \tag{4}$$

其中 m,n 分别代表卷积核窗口的高度和宽度,得到的卷积结果会作为激活函数 f 的输入,经过 f 处理后的结果为该层的输出特征图(feature maps),激活函数如公式(5)所示:

$$c_{i,j} = f(s(i,j) + b) \tag{5}$$

激活函数 f 常用的为 sigmoid 或 tanh 等非线性函数,b 为偏置项.

2) 池化层

池化层的任务是对卷积层的输出特征图进行下采样,简化卷积层的输出,从而减少训练参数,并且可以避免过拟合.常用的采样方法为最大值下采样(Max-Pooling)和平均值下采样(Mean-Pooling).假设采样窗口的宽度为 w,高度为 h,池化过程先以窗口的大小作为步长,把卷积层的输出特征矩阵划分为若干个 w × h 大小的子区域,在使用相关的采样方法对每个子区域进行采样.最大值下采样方法得到的是该区域的最大特征值,平均值下采样得到的是该区域所有特征的平均值.

3) 全连接层

全连接(FC)处于卷积神经网络最后,通过多层的卷积层与池化层处理后,原始的输入数据被映射到了隐含特征空间,全连接层则通过特征空间转换,把得到的"分布式特征表示"映射为样本的标记空间.全连接层可由卷积操作实现,使用 w * h 个 1 × 1 大小的卷积核对前层卷积输出进行卷积计算,w,h 分别代表前层卷积输出的特征图的宽和高.全连接给整个神经网络带来了大量参数,是限制算法性能的一个重要瓶颈,近期有学者提出了一些新的网络模型,如 ResNet 和 GoogLeNet 等[28],这些模型使用全局平均池化(global average pooling,GAP)取代 FC,来融合前层卷积得到的特征图,仍采用 softmax 等非线性损失函数作为网络目标函数引导学习过程.

3 卷积神经网络文本分类模型构建

3.1 改进型 TF-IDF 算法

TF-IDF 是一种计算词语权重的经典统计方法,由词频(term frequency,TF)和逆向文档频率(inverse document frequency,IDF)两部分数据组成,词频计算如公式(6)所示:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_{k=1}^{k} n_{k,j}} \tag{6}$$

$tf_{i,j}$ 代表词语 w_i 在文档 d_j 中出现频率,$n_{i,j}$ 为 w_i 在文档 d_j 中出现的次数,分母为文档 d_j 中所有词语出现次数总和,k 为文档 d_j 中不同词语的个数.逆向文档频率计算如公式(7)所示:

$$idf_i = \log \frac{n_d}{df(d, w_i) + 1} \tag{7}$$

其中 idf_i 代表词语 w_i 在文本库 d 中的逆向文档频率,n_d 为文本库 d 中文档的总个数,$df(d, w_i)$ 为文档库 d 中包含词语 w_i 的文档个数,加 1 是为了防止 $df(d, w_i)$ 为零的情况.最后 TF-IDF 归一化处理的计算如公式(8)所示:

$$tf-idf_{i,j} = \frac{tf_{i,j} \times idf_i}{\sqrt{\sum w_i \in d_j \left[tf_{i,j} \times idf_i\right]^2}} \tag{8}$$

通过公式可以看出,词语 w_i 对文档 d_j 的重要程度和它在文档 d_j 中出现的频率成正比,和在整个文本库 d 中包含词语 w_i 的文档数成反比.

在文本分类研究中,文本库中的文本通常被标记成几个不同类别,而 TF-IDF 算法只考虑特征词在整个文本库中出现的总频率,忽略了在类别中的分布,例如某个词语 w_i 在文本库中的几个类别的文本中出现频率较高,而在其他几个类别的文本中出现频率较低,说明该 w_i 对文本的判别具有一定贡献,而传统的 TF-IDF 算法没有考虑这种不同类别间的分布情况,导致某些对类别判断具有贡献的词丢失.因此本文提出引入类频方差的 TF-IDF 算法,类频方差衡量的是词语在不同类别的分布情况,计算如公式(9)所示:

$$\tau_i = \frac{\sqrt{\sum_{j=1}^{N} \left(\frac{df(d, w_i)}{N} - df(d_{c_j}, w_i)\right)^2}}{N} \tag{9}$$

τ_i 为词语 w_i 的类频方差,N 为文本类别数,$df(d, w_i)$ 为整个文本库 d 中包含词语 w_i 的文档个数,$df(c_j, w_i)$ 为在类别 c_j 中包含词语 w_i 的文档个数.τ_i 越大说明词语 w_i 在类别中波动越大,分布越不均匀,对类别的判断作用越大,所以基于类频方差的 TF-IDF 算法计算如公式(10)所示:

$$tf-idf-\tau_{i,j} = tf-idf_{i,j} * \tau_i \tag{10}$$

$tf-idf_{i,j}$ 和 τ_i 的计算分别如公式(8)、公式(9)所示.公式(10)就是本文提出的改进型 TF-IDF 算法.

3.2 CTMWT 模型构建

针对传统机器学习文本分类算法的语义特征表达弱、文本表示维度高、词序丢失、矩阵稀疏等问题,本文提出基于 Word2vec、改进型 TF-IDF 和卷积神经网络三者相结合的文本分类模型(CTMWT),如图 2 所示.

1) Word2vec 词向量库建设

Word2vec 词向量的获取一般有两种方式,一种是使用开源的全局词向量库,这种词向量库是通过对全网超大规模文本库训练得到,如 2018 年北京师范大学和人民大学的自然语言处理小组开源的一套中文词向量库.还有一种方式是根据自己收集的文本库训练,得到一个局部的词向量库.由于我们处理的是针对某个问题领域的文本分类,所以采用该领域的文本进行向量库的训练,得出的词向量更贴合该问题领域.具体词向量库建设流程如图 3 所示.

文本库包含所用的训练样本和测试样本,神经网络隐藏层神经元的个数为 100-300 之间,隐藏层神经元的个数即词向量的维数,针对大规模输入-隐藏层参数矩阵训练时梯度下降慢的问题采用负采样(negative sampling)技术改进.

2) 文本向量表示

先对文本d_i进行分词处理$W_i = [w_i, \cdots, w_n]$，n 为词语个

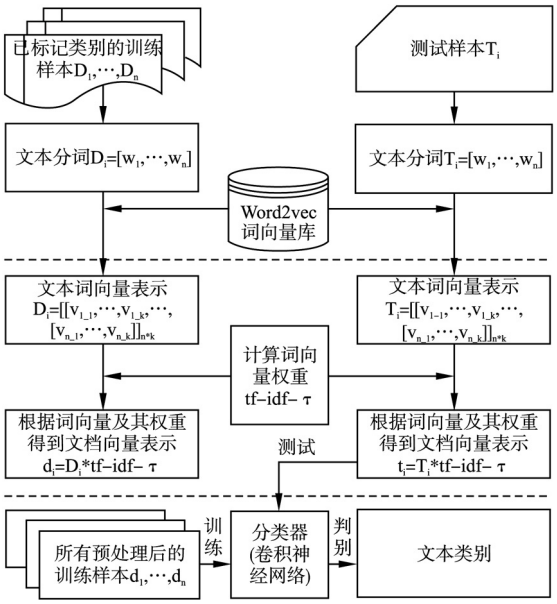

图 2　CTMWT 模型

Fig. 2　CTMWT model

数. 在根据 Word2vec 词向量库把分词后的文本替换成低维数值向量$VW_i = [V_{w_1}, \cdots, V_{w_n}]$，$V_{w_i}$ 为词w_i的词向量，$V_{w_i} = [v_1, v_2, \cdots, v_k]$，$k$ 为词向量的维度. 使文本表示从神经网络难处理的高纬度高稀疏传统数据，变成了类似图像的连续稠密

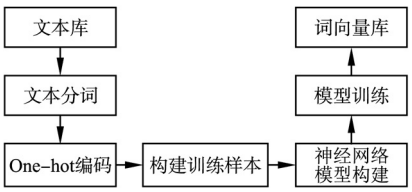

图 3　词向量库建设

Fig. 3　Construction of word vector library

矩阵数据表示. 并且这种文本表示免去了传统机器学习文本分类算法中人工特征选择的繁琐工作，让文本原始信息得到了最大程度保留. 但 Word2vec 的词向量不能刻画词语对文本的重要度，所以用公式(10) 改进型 TF-IDF 算法进行向量词权重计算，最终文本表示如公式(11) 所示:

$$vec(d_i) = \sum_{t \in W_i} V_t * \ tf - idf - \tau_{t,i} \tag{11}$$

其中 $tf - idf - \tau_{t,i}$ 的计算见公式(10).

3) 卷积神经网络文本分类

通过公式(11) 处理后的文本被表示成类似图像的连续稠密矩阵数据，深度学习算法具有很强的数据迁移性，在图像领域取得很好效果的卷积神经网络也可以迁移到文本处理领域，卷积神经网络文本分类模型如图 4 所示.

卷积网络一般是由多个卷积层、池化层连接组成，图 4 只是画出了一层. 一个卷积层中有多个不同的卷积核 w，$w \in R^{hk}$，h 为卷积核的高度，k 为词向量的空间维度，卷积核以步长 1 向下滑动，每经过一个文本向量 $h*\ k$ 的窗口时进行卷积运算，产生一个新的特征值，计算如公式(12) 所示:

$$c_i = f(w \cdot W_{i:i+h} + b) \tag{12}$$

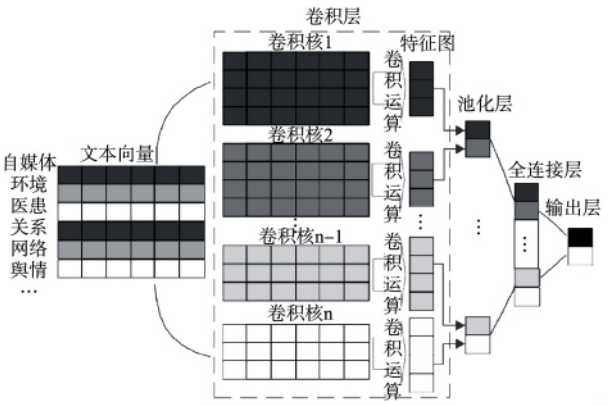

图 4　卷积神经网络文本分类模型

Fig. 4　Text classification model based on convolutional neural network

$W_{i:i+h}$ 为一个长度为 h 的词语序列$(W_i, W_{i+1}, \cdots, W_{i+h})$，$w$ 为卷积核矩阵权重参数，b 为偏置项，$b \in R$，操作符$(\cdot)$ 为卷积计算，f 为激活函数. 一个卷积核对文本向量处理后得到一个特征图 $c = (c_1, c_2, \cdots, c_{n-h+1})$，$n$ 为文本中词语的个数. 池化层使用 1-max-pooling 对特征图的特征进行提取，$\hat{c} = max\{c\}$，通过池化层处理后，不同长度的文本都变成了相同长度的特征表示了. 全连接层的输入为池化层的特征输出，输入为 $v = (\hat{c}_{1,1}, \cdots, \hat{c}_{1,q}, \cdots, \hat{c}_{2,1}, \cdots, \hat{c}_{2,q}, \cdots, \hat{c}_{p,1}, \cdots, \hat{c}_{p,q})$，p 为卷积核的种类，q 为每种卷积核的个数. 输出层使用 softmax 函数进行类别判定，图 4 中显示的是二分类情况，但也可以是多分类.

4　实验分析

4.1　实验数据

实验环境主要基于 TensorFlow 和 Python3. 6. 实验数据来源于清华大学自然语言处理实验室的中文文本分类数据集 THUCNews，该数据集根据新浪新闻 RSS 订阅频道 2005-2011 年间的历史数据筛选过滤生产，共包括 14 个新闻类别，共 74 万篇新闻文档，均为 UTF-8 的纯文本格式. 本实验为了减少神经网络的训练时间，从 THUCNews 数据集中选取了体育、财经、房产、家居、教育、科技、时尚、时政、游戏、娱乐 10 个类别，每个类别 12000 篇新闻文档，其中 10000 为训练集(train set)、1000 为测试集(test set)、1000 为验证集(validation set)，验证集用于指导神经网络结构的调整，总共训练集 10 万，测试集 1 万，验证集 1 万. 实验的第一步是要得到词向量库，使用所有的训练集、测试集、验证集共 12 万文本进行训练，对 12 万的文本分词后，得到近 5000 万的词汇，包含 60 万个不同的词，选取大小为 4 的上下文窗口，得到(输入词，输出词) 的训练样本近 2 亿组，设定词向量维度为 100，采用 CBOW 模型进行训练得到词向量库.

4.2　CTMWT 模型实验

4.2.1　CTMWT 模型实验

卷积神经网搭建参数的不同也会影响到实验效果，但这些不同参数的实验对比不是本文重点，所以通过查阅相关资

料,确定本实验的卷积神经网络的主要参数如表 1 所示.

表 1 卷积神经网络参数

Table 1 Convolution neural network parameters

参数名称	参数值
词向量维度	100
卷积核个数	256
卷积核窗口高度	3,4,5
池化方法	1-max pooling
全连接层神经元个数	128
每批训练大小(batch_size)	64
迭代轮次(num_epochs)	10
丢弃率(dropout_keep_prob)	0.5
学习率	1e-3

通过表 1 参数搭建卷积神经网络,采用训练集(train set)对 CTMWT 模型进行训练,得出分类结果模型;再使用测试集(test set)对分类结果模型进行性能测试,分类算法的性能主要从精准率(precision)、召回率(recall)、F1-Measure 三个指标进行评价,测试结果如表 2 所示.

表 2 算法测试结果

Table 2 Algorithm test results

类别	精准率(%)	召回率(%)	F1-Measure(%)
体育	99.0	99.5	99.2
财经	95.1	99.1	97.1
房产	99.9	99.6	99.7
家居	98.0	87.8	92.6
教育	90.8	92.3	91.5
科技	91.7	98.7	95.1
时尚	97.7	96.5	97.1
时政	93.8	95.7	94.8
游戏	98.4	96.7	97.5
娱乐	98.8	96.4	97.6
平均值	96.3	96.2	96.2

通过测试结果发现,平均精准率、召回率、F1-Measure 分别达到了 96.3%、96.2%、96.2%.

4.2.2 CTMWT 模型与传统机器学习文本分类算法实验对比分析

为了进一步验证 CTMWT 模型的有效性,分别选取朴素贝叶斯(NB)、K 最近邻(KNN)、支持向量机(SVM)三类传统机器学习文本分类算法进行实验对比,这三类传统机器学习文本分类算法使用信息增益(IG)进行特征选择、使用 TF-IDF 进行特征权重计算、使用向量空间模型进行文本表示.数据集依然使用 4.1 节介绍的试验数据,性能评价使用 10 个类别的平均 F1-Measure 指标,实验结果图 5 所示.

通过实验发现 CTMWT 模型的性能明显优于传统机器学习算法.为了进步和传统机器学习文本分类算法对比,在分别选择不同数量的训练数据进行实验,训练数据从每个类别选取的数量从 1000 到 10000 之间递增,每次增加 1000,共 10 次对比.实验结果如图 6 所示.

通过图 6 实验结果发现在训练数据集比较小的情况下,CTMWT 模型相比传统的机器学习算法优势并不明显,只有当训练数据达到一定规模后优势才慢慢体现,这是因为相比传统机器学习算法,卷积神经网络中的学习参数更复杂,只有

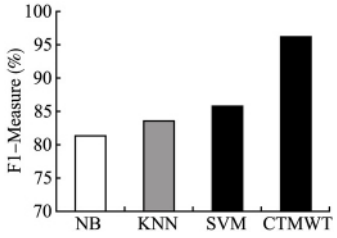

图 5 算法性能对比

Fig. 5 Algorithm performance comparison

在大量训练样本的情况下才能获得较好的学习效果.并且发现 CTMWT 模型随着训练数据的不断增加,学习能力也随之增加,而传统机器学习算法在训练样本达到一定规模后,随着训练数据的增加,算法的学习能力并没有得到明显的提升,这也是传统机器学习算法的一个弱点.

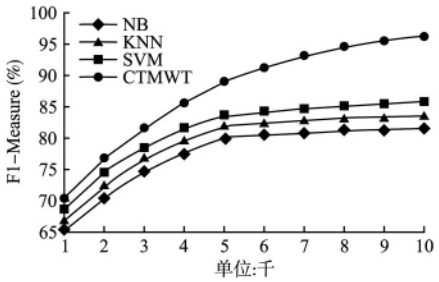

图 6 不同训练集大小的性能对比

Fig. 6 Performance comparison of algorithms with different training set sizes

4.2.3 不同词向量权重的 CTMWT 模型实验对比分析

为了验证改进型 TF-IDF 词向量权重算法的有效性,分别对不加词向量权重(NONE)、基于传统 TF-IDF 计算词向量权重(TF-IDF)和本文提出的权重计算方法(TF-IDF-τ)进行 CTMWT 模型实验对比.实验结果如图 7 所示.

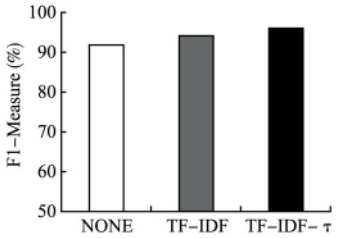

图 7 不同词向量权重的 CTMWT 模型性能对比

Fig. 7 Performance comparison of CTMWT models with differ vector weights

通过实验发现,引入词向量权重后,算法的性能有所提高,因为引入权重是对词向量特征的增强.本文基于类频方差改进 TF-IDF 算法又比传统的 TF-IDF 算法性能高,所以证明本文引入的词向量权重计算方法是有效的.并且通过实验发现,就算不引入任何的词向量权重,基于卷积神经网络的文本分类算法依然保持了较高的分类性能,这一方面得益于基于

词向量的文本表示最大程度的保留了文本原始信息,另一方面得益于卷积神经网络强大的学习能力.

5 总　结

随着信息化和互联网的快速发展,产出了海量的文本数据,如何对这些文本进行准确有效的分类一直是研究的热点.本文针对传统机器学习文本分类算法语义特征表达弱、文本表示维度高、词序丢失、矩阵稀疏导致分类效果不佳的问题,提出基于 Word2vec、改进型 TF-IDF 和卷积神经网络三者相结合的文本分类模型(CTMWT),使用 Word2vec 进行词语表示,生产具有语义表达能力的低维稠密性词向量;引入类频方差改进型 TF-IDF 算法,计算每个词向量在文本中的权重,构建基于词向量和权重的向量文本表示,保留词语上下文关系;最后利用使用卷积神经网络强大学习能力,对大量文本向量进行深度学习.通过实验发现 CTMWT 模型不仅能实现文本的准确分类,而且相比于传统机器学习文本分类算法具有更好的分类性能.CTMWT 模型免去了传统机器学习文本分类算法中人工特征选择的繁琐工作,让文本原始信息得到了最大程度保留,为卷积神经网络的学习提供了良好的基础,但在实验过程中也发现 CTMWT 模型训练时间复杂度远高于传统机器学习文本分类算法,这是深度学习不可避免的问题,也是相关学者继续深入研究的方向.

References:

[1] Glavaš G, Nanni F, Ponzetto S P. Unsupervised text segmentation using semantic relatedness graphs [C]. Joint Conference on Lexical and Computational Semantics, 2016: 125-130.

[2] Lample G, Ballesteros M, Subramanian S, et al. Neural architectures for named entity recognition [C]. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Stroudsburg: Association for Computational Linguistics, 2016: 260-270.

[3] Lu Y, Liang M, Ye Z, et al. Improved particle swarm optimization algorithm and its application in text feature selection [J]. Applied Soft Computing, 2015, 35(C): 629-636.

[4] Nalluri S P, Kurra R R. Feature selection based on term frequency and term document frequency for text clustering [J]. International Journal of Applied Engineering Research, 2015, 10(10): 26175-26190.

[5] Shang C, Li M, Feng S, et al. Feature selection via maximizing global information gain for text classification [J]. Knowledge-Based Systems, 2013, 54(4): 298-309.

[6] Pandis N. The chi-square test [J]. American Journal of Orthodontics & Dentofacial Orthopedics, 2016, 150(5): 898-899.

[7] Zhao Jing, Shao Xiong-kai, Liu Jian-zhou, et al. Study on feature selection method in text classification [J/OL]. Application Research of Computers, 2019, (8): 1-8. http://kns.cnki.net/kcms/detail/51.1196.TP.20180424.1022.034.html, 2018-10-05.

[8] Li Ming-jiang. Research on method of feature selection in text combined with word frequency in class [J]. Application Research of Computers, 2014, 31(7): 2024-2026.

[9] Chen Lei, Li Jun. Text feature selection methods based on word vector [J]. Journal of Chinese Computer Systems, 2018, 39(5): 991-994.

[10] Zhu L, Wang G, Zou X. Improved information gain feature selection method for Chinese text classification based on word embedding [C]. International Conference on Software and Computer Applications, ACM, 2017: 72-76.

[11] Zhang D, Xu H, Su Z, et al. Chinese comments sentiment classification based on word2vec and SVM perf [J]. Expert Systems with Applications, 2015, 42(4): 1857-1863.

[12] Meng J, Lin H, Li Y. Knowledge transfer based on feature representation mapping for text classification [J]. Expert Systems with Applications, 2011, 38(8): 10562-10567.

[13] Poli R, Healy M, Kameas A. Theory and applications of ontology: computer applications [M]. Dordrecht: Springer, 2010: 231-243.

[14] Shi Jie, Zhou Lan-jiang, Xian Yan-tuan, et al. Chinese-thai cross-language text similarity computing based on WordNet [J]. Journal of Chinese Information Processing, 2016, 30(4): 65-70.

[15] Dittinger E, Barbaroux M, D´Imperio M, et al. Professional music training and novel word learning: from faster semantic encoding to longer-lasting word representations [J]. J Cogn Neurosci, 2016, 28(10): 1584-1602.

[16] Chen K, Zhang Z, Long J, et al. Turning from TF-IDF to TF-IGM for term weighting in text classification [J]. Expert Systems with Applications, 2016, 66(C): 245-260.

[17] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality [C]. International Conference on Neural Information Processing Systems, Curran Associates Inc, 2013: 3111-3119.

[18] Li Xiao, Xie Hui, Li Li-jie. Research on sentence semantic similarity calculation based on Word2vec [J]. Computer Science, 2017, 44(9): 256-260.

[19] Zhang Y, Jin R, Zhou Z H. Understanding bag-of-words model: a statistical framework [J]. International Journal of Machine Learning & Cybernetics, 2010, 1(1-4): 43-52.

[20] Jing L, Ng M K, Huang J Z. Knowledge-based vector space model for text clustering [J]. Knowledge & Information Systems, 2010, 25(1): 35-55.

[21] He Ming, Sun Jian-jun, Cheng Ying. Text classification based on naive Bayes: a review [J]. Information Science, 2016, 34(7): 147-154.

[22] Zhou Qing-ping, Tan Chang-geng, Wang Hong-jun, et al. Improved KNN text classification algorithm based on clustering [J]. Application Research of Computers, 2016, 33(11): 3374-3377+3382.

[23] Zhang Hua-xin, Pang Jian-gang. Research on text classification based on SVM and KNN [J]. Journal of Modern Information, 2015, 35(5): 73-77.

[24] Zeng Shui-fei, Zhang Xiao-yan, Du Xiao-feng, et al. New method of text representation model based on neural network [J]. Journal on Communications, 2017, 38(4): 86-98.

[25] Zhang Qian, Gao Zhang-min, Liu Jia-yong. Research of weibo short text classification based on Word2vec [J]. Netinfo Security, 2017, (1): 57-62.

[26] Zhang Qun, Wang Hong-jun, Wang Lun-wen. Classifying short texts with word embedding and LDA model [J]. Data Analysis and Knowledge Discovery, 2016, (12): 27-35.

[27] Lu Shu-bao, Wang Ming-yue, Zhai Xiang, et al. AN information text classification algorithm based on DBN [J]. Journal of Harbin University of Science and Technology, 2017, 22(2): 105-111.

[28] Khan R U, Zhang X, Kumar R. Analysis of ResNet and GoogleNet models for malware detection [J]. Journal of Computer Virology & Hacking Techniques, https://doi.org/10.1007/s11416-018-0324-z: 1-9.

附中文参考文献:

[7] 赵 婧,邵雄凯,刘建舟,等. 文本分类中一种特征选择方法研究[J/OL]. 计算机应用研究,2019,(8):1-8. http://kns.cnki.net/kcms/detail/51.1196.TP.20180424.1022.034.html,2018-10-05.

[8] 李明江. 结合类词频的文本特征选择方法的研究 [J]. 计算机应用研究,2014,31(7):2024-2026.

[9] 陈 磊,李 俊. 基于词向量的文本特征选择方法研究 [J]. 小型微型计算机系统,2018,39(5):991-994.

[14] 石 杰,周兰江,线岩团,等. 基于 WordNet 的中泰文跨语言文本相似度计算 [J]. 中文信息学报,2016,30(4):65-70.

[18] 李 晓,解 辉,李立杰. 基于 Word2vec 的句子语义相似度计算研究 [J]. 计算机科学,2017,44(9):256-260.

[21] 贺 鸣,孙建军,成 颖. 基于朴素贝叶斯的文本分类研究综述 [J]. 情报科学,2016,34(7):147-154.

[22] 周庆平,谭长庚,王宏君,等. 基于聚类改进的 KNN 文本分类算法 [J]. 计算机应用研究,2016,33(11):3374-3377+3382.

[23] 张华鑫,庞建刚. 基于 SVM 和 KNN 的文本分类研究 [J]. 现代情报,2015,35(5):73-77.

[24] 曾诚飞,张笑燕,杜晓峰,等. 基于神经网络的文本表示模型新方法 [J]. 通信学报,2017,38(4):86-98.

[25] 张 谦,高章敏,刘嘉勇. 基于 Word2vec 的微博短文本分类研究 [J]. 信息网络安全,2017,(1):57-62.

[26] 张 群,王红军,王伦文. 词向量与 LDA 相融合的短文本分类方法 [J]. 现代图书情报技术,2016,(12):27-35.

[27] 吕淑宝,王明月,翟 祥,等. 一种深度学习的信息文本分类算法 [J]. 哈尔滨理工大学学报,2017,22(2):105-111.