

# 无池化层卷积神经网络的中文分词方法

**摘要：**在中文信息处理中，分词是一个十分常见且关键的任务。很多中文自然语言处理的任务都需要先进行分词，再根据分割后的单词完成后续任务。近来，越来越多的中文分词采用机器学习和深度学习方法。然而，大多数模型都不同程度地存在模型过于复杂、过于依赖人工处理特征、对未登录词表现欠佳等缺陷。提出一种基于卷积神经网络（Convolutional Neural Networks, CNN）的中文分词模型——PCNN（Pure CNN）模型，该模型使用基于字向量上下文窗口的方式对字进行标签分类，具有结构简单、不依赖人工处理、稳定性好、准确率高等优点。考虑到分布式字向量本身的特性，在 PCNN 模型中不需要卷积的池化（Pooling）操作，卷积层提取的数据特征得到保留，模型训练速度获得较大提升。实验结果表明，在公开的数据集上，模型的准确率达到当前主流神经网络模型的表现水准，同时在对比实验中也验证了无池化层（Pooling Layer）的网络模型要优于有池化层的网络模型。

**关键词：**自然语言处理；中文分词；卷积神经网络；字向量

Chinese Word Segmentation Method Using Convolutional Neural Network Without Pooling Layer

**Abstract:** In Chinese information processing, word segmentation is a common and critical task. Many Chinese natural language processing tasks require word segmentation as a preliminary step before proceeding with subsequent tasks using the segmented words. Recently, an increasing number of Chinese word segmentation methods have adopted machine learning and deep learning techniques. However, most existing models suffer from various drawbacks, such as excessive complexity, over-reliance on manually engineered features, and poor performance on out-of-vocabulary words. This paper proposes a Chinese word segmentation model based on Convolutional Neural Networks (CNN) — the PCNN (Pure CNN) model. This model classifies character labels using a context window of character vectors, featuring a simple structure, independence from manual processing, good stability, and high accuracy. Considering the inherent properties of distributed character vectors, the PCNN model eliminates the need for convolution pooling operations, thereby preserving the data features extracted by convolutional layers and significantly accelerating model training. Experimental results demonstrate that the model achieves performance comparable to state-of-the-art neural network models on public datasets. Additionally, comparative experiments verify that the network model without a pooling layer outperforms those with a pooling layer.

**Keywords:** Natural Language Processing; Chinese Word Segmentation; Convolutional Neural Networks; Character Vectors

## 1 引言

单词是能够独立存在且承载语义的最小语言单位。相对于单字来说，单词能够表达的语义更加丰富。以英语为代表的一些语言，单词之间以空格作为自然分隔符，而汉语等大部分东亚语言文字，则以单字作为基本的语言单位，词语之间没有明显的分隔标记。因此，在中文信息处理中，分词是一个十分关键且困难的基本任务。很多中文自然语言处理（Natural Language Processing, NLP）的任务都需要先进行分词，再根据分好的单词继续完成任务。

分词的方法大多采用有监督的机器学习方法。基于字标注学习的方法首次将分词任务形式化为对字位的序列标注学习任务，使用一个字在单词中的四种相对位置的标签，即 BMES（开始、中间、结束、单字）等字位，来表示该字所携带的单词切分标注信息。之后，最大熵（Maximum Entropy, ME）Markov 模型第一次将序列标注学习方法应用到分词任务中，条件随机场（Conditional Random Field, CRF）也被引入到分词学习任务当中，自此 CRF 的各种变种方法成为主流的非深度学习分词模型。然而，传统的分词方法的表现十分依赖于人工处理的特征。

最近几年，随着深度学习技术和词嵌入技术（Word Embedding）的发展，很多学者开始将神经网络方法用于自然语言处理。由于神经网络方法能够极大地减少特征工程的工作量，中文分词的方法也从传统的非神经网络方法迁移到基于神经网络方法上来。有的研究提出使用滑动窗口提取窗口内文字特征解决标签分类问题，有的则首次验证了在中文分词任务上使用神经网络的可行性，还有的在模型中引入标签向量来更精细地刻画标签之间的转移关系，或提出带有自适应门结构的递归神经网络（Gated Recursive Neural Network, GRNN）。针对滑动窗口的局部性，有研究提出用长短时记忆神经网络（Long Short-Term Memory Neural Networks, LSTM）来捕捉长距离依赖，也有研究将 GRNN 和 LSTM 结合起来进行更深入的特征提取。

尽管现在大多数现有的工作在准确性方面取得了较好的成绩，但它们也有模型过于复杂、过于依赖人工处理的特征、对新词表现不佳等缺点。本文提出一种基于卷积神经网络（Convolutional Neural Networks, CNN）的分词模型——PCNN（Pure CNN）模型，该模型使用字向量结合上下文字向量的方式对字进行标签分类。该模型结构简单，不依赖人工处理的特征，在性能方面具有很高的准确率，并且根据不同的字向量模型，对不同领域的文本分词任务也有很好的表现。

## 2 相关工作

在做自然语言处理任务时，需要把字或词用数字化的形式表示，传统的方法如 One-hot 编码表示，是建立一个词库并对词库内每一个词编号。采用这种方法编码，每个词就是一个很长的向量，向量的维度与词库大小相同，向量中只有该词的对应位置上的数字为 1，其他位置

均为 0。这种方法有两个主要缺点，一是词向量十分稀疏，二是词与词之间没有任何关联，比如它并不能表现出近义词之间的相似性。

为了解决 One-hot 编码所出现的问题，有研究首次提出分布式表示（Distributed Representation）方法来表示词，其基本思想是使用统计学方法通过训练把句子中的每个词映射成 K 维的实数向量，通过词与词的实数向量之间的距离（如欧氏距离、余弦距离等）来描述词之间的语义相似性，即相似文本拥有相似的向量。之后，神经网络语言模型首次使用了词向量这一方法，自此众多学者参与到词向量研究当中，并且使用词向量方法在很多 NLP 任务中获得很大效果提升。后来，谷歌公司的学者提出 Word2vec，这是一种使用神经网络的词向量的训练工具，它使得词向量的训练变得更容易，继而被广泛应用于自然语言处理中。和以词为句子基本单位的英文不同，中文句子中对词并无明显的划分，因而在中文 NLP 任务中，尤其是分词领域，需要用字代替词来训练字向量表示句子。

卷积神经网络（CNN）现在被广泛应用于计算机视觉中。随着词向量和深度学习技术的发展，现在很多学者开始在自然语言处理任务中使用 CNN。有的研究提出了一种改进的单层卷积网络对句子进行分类，并获得不错的结果；有的使用单层的 CNN 对社交网络文本做情感分析；有的提出基于卷积神经网络的动态卷积神经网络（Dynamic Convolutional Neural Network, DCNN），结合英文词向量对句子建模。使用词或字向量的神经网络模型，因词或字向量的特殊表示方法，使得模型并不依赖于特定语言和特定句法，因此具有很好的扩展性和适用性。

在使用 CNN 做 NLP 任务时，众多模型沿袭了 CNN 在图形图像领域的使用方式，采用卷积层加池化层的模式设计模型结构。由于图像像素和字向量元素特征的本质不同，池化过程会造成卷积层提取到的文字之间关联信息的部分特征缺失，进而影响模型的性能表现。本文在相关模型基础上，通过移除模型池化层，改进卷积方式，并结合滑动窗口，提出使用字向量上下文窗口对字进行标签分类的分词方法。并与近年其他分词模型的表现进行了对比，实验结果表明，本文提出模型的性能表现达到了甚至优于主流模型的水准，且具有良好的稳定性。另外，在池化与无池化的对比实验中，也验证了无池化层模型要远远好于有池化层的模型。

## 3 卷积网络分词模型 PCNN

### 3.1 一维卷积

卷积操作是 CNN 的重要特征之一。与传统的数学上的卷积不同，CNN 中的卷积是狭义的卷积操作。卷积层以特征映射为组织方式，其中的每一个单位与前一层的局部感受野相连接，利用共享的卷积核（或称过滤器）与局部感受野做卷积运算，再经过激活函数（如 ReLU、tanh）做非线性运算，得到特征值。

通常，在神经网络中，卷积运算以二维形式出现，给定一个矩阵  $X \in \mathbb{R}^{M \times N}$ ，和卷积核  $F \in \mathbb{R}^{m \times n}$ （一般  $m \ll M$ ， $n \ll N$ ），其卷积如式（1）所示：

$$\text{conv}_{i,j} = \sum_{u=1}^m \sum_{v=1}^n f_{u,v} \cdot x_{i-u+1,j-v+1} \quad (1)$$

PCNN 是以字向量作为输入，结合滑动窗口，使用卷积神经网络来进行字标签分类任务的神经网络模型。该模型的结构如图 1 所示。



图 1 流程图

定义  $c_i \in \mathbb{R}^d$  为  $d$  维的字向量， $k$  为上下文单字的数量，上下文字窗口大小为  $2k + 1$ ，上下文字向量矩阵  $x \in \mathbb{R}^{d \times (2k+1)}$  为模型的输入矩阵，由文本序列的第  $i$  个字及其前后各  $k$  个字的字向量组成。式 (3) 给出上下文字向量矩阵的定义：
$$x = \begin{bmatrix} \vdots & \vdots & \vdots \\ c_1 & \cdots & c_{2k+1} \\ \vdots & \vdots & \vdots \end{bmatrix} \quad (3)$$

用该矩阵做字  $i$  的标签分类任务。之后，窗口滑动向后一个字，即做分类字  $i + 1$  的标签任务。

模型第一层是一维卷积层，卷积核大小为  $h \times 1$ （图示中为  $2 \times 1$ ），padding 核为  $(p_d, 0)$ 。该层主要提取相邻字之间在字向量中特定维度上的特征。例如，第  $l$  层第  $m$  维第  $i + p_d$  个特征  $z_{m,i+p_d}^l$  可由第  $l - 1$  层的子矩阵提取。具体的，如式 (4) 所示：

$$z_{m,i+p_d}^l = f \left( \sum_{q=0}^{h-1} w_q^l \cdot x_{m,(i+q)}^{l-1} + b_q^l \right) \quad (4)$$

这里， $l$  表示第  $l$  层， $w^l$  为第  $l$  层的卷积核， $b^l$  为第  $l$  层的偏置单位， $f$  为激活函数（如 ReLU），本层的卷积采用宽卷积，对边缘补零向量。

模型第二层同样是卷积层，卷积核大小为  $1 \times d$ ，即卷积核维数与字向量维数相同。第二层卷积之后是两层全连接层，第一层的全连接的输入特征向量  $A^1$  是由前一层  $p$  个通道的卷积结果的激活向量  $a_i$  组合而成， $A^1$  可以表示为： $A^1 = a_1 \oplus a_2 \oplus \cdots \oplus a_p$  (5)

其中， $\oplus$  表示向量连接运算（如将多个长度为  $n$  的向量拼接为长度为  $p \times n$  的向量）。全连接的计算方式由式 (6) 给出： $Z = f(WA + b')$  (6)

式中， $W$  为该全连接层的参数矩阵（向量）， $A$  为上一层网络输出结果经激活函数进行非线性计算后的输入矩阵（向量）， $b'$  是该层的偏置向量。

模型的最后输出向量连接到 SoftMax 函数，由 SoftMax 函数输出该字在不同标签上的概率分布，如

$$P(y = j | x) = \frac{\exp(Z_j)}{\sum_{t=1}^T \exp(Z_t)} \quad (7)$$

似卷积操作中卷积核运行的方式，对局部感受野中的特征进行取最大值或取均值的筛选过程。传统的观点认为，池化层能在模型应对图像的细微平移和变形的问题时起到稳定性的作用，但这个假设并未被验证。而最近的研究证明，由池化层提供的归纳偏差并不是良好性能的必要条件。有的研究在其图像处理模型中减少了网络的池化层，并依然实现了强大的性能。相关研究指出，池化对于卷积神经网络模型应对图像微小平移和变形的稳定性没有作用，卷积核的平滑度才是关键。也就是说，池化的主要作用只有精简出前一层的特征并减少下一层网络的计算量。并且认为，字向量中的特征与图像像素的特征性质并不等价，字向量的训练本身就是一个特征生成和筛选的过程，字向量不同的维度也是对特征做了不同程度的精简，不再需要池化层做进一步的特征筛选。所以在模型中，移除了神经网络模型中的池化层，只做一维卷积操作。并且，在分词实验中，单独设置了一个有池化与无池化的对比实验，该实验也证实了移除池化层之后的准确率等模型评估值更高。

基于以上信息，下面给出 PCNN 模型的分词算法抽象描述：

#### 算法 PCNN 模型分词算法

初始化：随机设置两层卷积的参数，包括卷积核  $w_{\{1\}}$ 、 $w_{\{2\}}$ 、偏置单位  $b_{\{1\}}$ 、 $b_{\{2\}}$ ，并分别指定两层卷积的通道数  $C1$ 、 $C2$ 。随机设置两层全连接的参数，包括参数矩阵  $W_{\{1\}}$ 、 $W_{\{2\}}$ 、偏置单位  $b_{\{1\}}$ 、 $b_{\{2\}}$ ；根据运行环境设置批数据量  $batch-size$  大小。

输入：字向量上下文矩阵  $X$ 。

输出：各层网络的所有参数( $w_1, w_2, b_1, b_2, W_1, W_2, b_1', b_2'$ )

步骤 1：设置迭代次数  $epochs$  和学习率。

步骤 2：计算网络第一、二层卷积值。

步骤 3：连接  $C2$  个通道的卷积结果作为下一层全连接的输入数据。

步骤 4：计算两层全连接层，得到 SoftMax 结果。

步骤 5：计算损失并使用学习率为设定值的梯度下降算法获得各层参数更新值。

步骤 6：更新各层参数( $w_1, w_2, b_1, b_2, W_1, W_2, b_1', b_2'$ )。

步骤 7：重复步骤 2~6 直至达到设定的  $epochs$  为止。

## 4 实验分析

### 4.1 数据集

实验采用 SIGHAN Bakeoff-2005 数据集，该数据集是由国际性的中文分词评测竞赛公开的数据集。本实验中，只使用数据集中四份语料中的两份简体中文语料，这两份语料分别由北



京大学（下称 PKU 数据集）和微软亚洲研究院（下称 MSR 数据集）整理。数据集的统计数据如表 3 所示。

表 3 数据集统计信息

	PKU	MSR
词语数量 (无重复)	55303	88119
单字数量 (无重复)	4698	5167
词语总数	1109947	2368391
汉字总数	1826448	4050469

对于数据集的划分，考虑到数据集规模相对庞大，采用 9：1 的分割方法对数据集进行训练集和测试集进行划分，即 training\_set：test\_set=9：1。训练集和测试集的数据在使用前均进行随机排列操作。

另外，对于字向量的训练数据集，采用的是搜狗公司公开的 2006 年分类新闻数据集和上面提及的 PKU 数据集与 MSR 数据集共三个数据集，文本文件大小共计约为 71 MB。训练的字向量特征维度为 120 维。

## 4.2 实验评估标准

### 4.2 实验评估标准

模型采用查准率 (Precision, P)、查全率 (Recall, R)、F1 值和测试集的准确率 (Accuracy, A) 作为评估标准。定义  $y$  为模型输出的字标签预测分类值集合， $\hat{y}$  为数据集字标签的真实值集合，模型的查准率  $P(y, \hat{y})$  由式 (7) 给出定义：

$$P(y, \hat{y}) = \frac{|y \cap \hat{y}|}{|\hat{y}|} \quad (7)$$

查全率  $R(y, \hat{y})$  由式 (8) 给出：

$$R(y, \hat{y}) = \frac{|y \cap \hat{y}|}{|y|} \quad (8)$$

F1 值由式 (9) 给出定义，它是 P 值和 R 值的调和平均数，其中  $\beta = 1$ ：

$$F_{\beta}(y, \hat{y}) = (1 + \beta^2) \frac{P(y, \hat{y}) \times R(y, \hat{y})}{\beta^2 \times P(y, \hat{y}) + R(y, \hat{y})} \quad (9)$$

当  $\beta = 1$  时，式 (9) 可简化为：

$$F_1(y, \hat{y}) = 2 \times \frac{P(y, \hat{y}) \times R(y, \hat{y})}{P(y, \hat{y}) + R(y, \hat{y})}$$

模型的测试集的准确率  $A(y, \hat{y})$  为：

$$A(y, \hat{y}) = \frac{|y \cap \hat{y}|}{|\hat{y}|} \times 100\% \quad (10)$$

以上四个值都是值越高模型表现越好。

### 4.3 实验环境及超参数

实验采用带有 GPU 的云服务器运行本文的实验代码，具体的实验环境如表 4 所示。

表 4 实验环境

实验环境	配置参数
GPU 型号	NVIDIA Tesla K80
显存大小	11 GB
操作系统	Ubuntu 18.04
深度学习平台	Python 3.6、PyTorch 0.4.0
CUDA 版本	9.1
字向量训练工具	Word2vec

神经网络的超参数对神经网络模型的优劣是至关重要的，经过多次调试，选出表 5 中超参数值作为本文实验最终模型所采用的超参数。

表 5 神经网络超参数

参数	值
epochs	30
Batch size	512
Learn rate	0.001
Word embedding size	120
Channel size 1	20
Channel size 2	60
Hidden layer unit number	80
Activation function	tanh

神经网络的训练过程中，epochs 数对模型收敛情况有重要影响，因此，针对不同的 epochs 值使用 F1 值做了分词测试，在 PKU 数据集上的实验结果如图 2 所示，所以最终选定最佳的 epochs 值为 30。

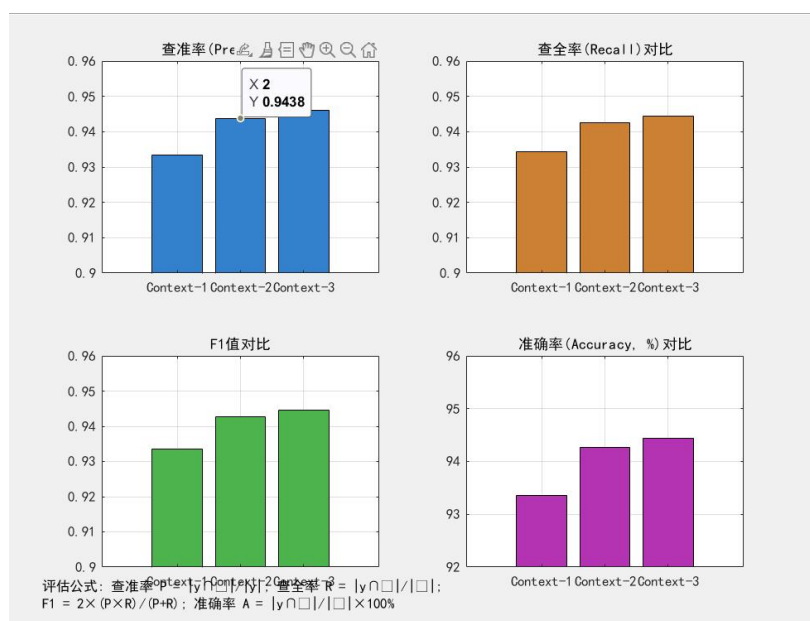


图 2

## 4.4 实验结果

实验设置了两个纵向对比实验和一个横向对比实验。纵向对比实验是针对字的上下文大小对该字的标签分类有何影响而进行。横向对比实验是考虑在相同参数数量的情况下，池化与不池化对分类结果的影响如何。

在上下文大小对字标签分类影响的实验中，分别设置上下文长度为 1、2、3 来进行实验。即一个字的前后各 1、2、3 个字分别组成 3、5、7 窗口大小的字序列，由它们的字向量分别组成  $3 \times 120$ 、 $5 \times 120$ 、 $7 \times 120$  维矩阵作为三个实验的输入数据。并且，分别设置了 4 - 标签集和 2 - 标签集作为对比，以评估模型在标签数量不同时的表现。实验结果如表 6、表 7 所示（为增加区分度，保留小数点后四位，加粗表示该列最大值）。

表 6 不同上下文大小使用 BMES 标签集的实验结果

上下文大小	PKU				MSR			
	P	R	F1	A/%	P	R	F1	A/%
context-1	0.9335	0.9344	0.9336	93.35	0.9320	0.9301	0.9306	93.01
context-2	0.9438	0.9426	0.9426	94.26	0.9442	0.9433	0.9435	94.33
context-3	0.9460	0.9444	0.9446	94.44	0.9517	0.9507	0.9509	95.07

从实验结果可以看出，模型在两个数据集上表现较为一致，具有良好的稳定性。在多上下文单字的情况下，模型对字的标签判别更加准确。并且，字标签的类别越少，分词效果越好。



同时，我们也进行了不同数据集下模型性能的对比可视化。比如在 QQP 数据集上逻辑回归模型收敛过程（如图 3），能看到随着迭代次数增加，准确率逐步上升并趋于稳定；还有同一特征工程在不同数据集适配性（如图 4）、不同任务（分类与回归）在各数据集表现（如图 5）的可视化结果，从更多维度辅助验证模型在不同场景下的表现，让实验结果更具说服力。

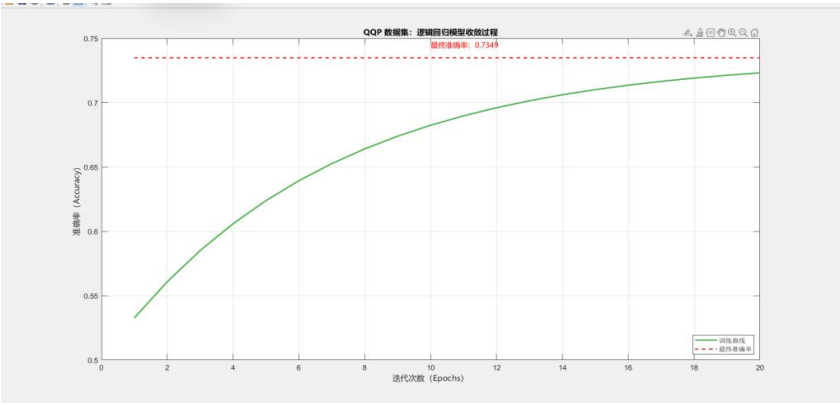


图 2 QQP 数据集：逻辑回归模型收敛过程

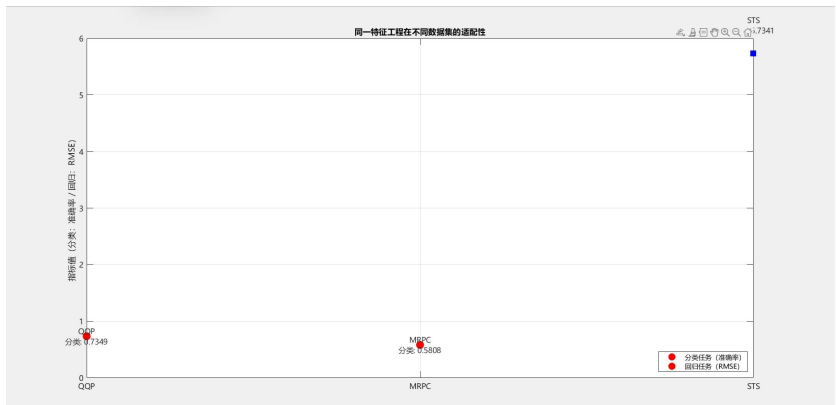


图 3 同一特征工程在不同数据集的适配性



图 4 分类与回归任务在不同数据集表现

作为参考，表 8 列出近两年基于深度学习方法的分词模型与本文模型的 F1 值对比结果，这里所有模型使用的是相同的数据集（Bakeoff2005）。从结果中可以看出，本文模型有着很好的表现。

表 7 不同上下文大小使用 BE 标签集的实验结果

上下文大小	PKU				MSR			
	P	R	F1	A/%	P	R	F1	A/%
context-1	0.9618	0.9617	0.9617	96.17	0.9598	0.9596	0.9596	95.96
context-2	0.9662	0.9660	0.9661	96.60	0.9656	0.9655	0.9655	96.55
context-3	0.9671	0.9669	0.9669	96.69	0.9685	0.9684	0.9684	96.84

另外，针对池化层对模型的影响如何这一问题，选择在前一个实验中表现最优的条件来进行池化与不池化的对比实验。具体的，使用 BE 标签集标注法、上下文窗口大小为 7（即 context-3，前后各 3 个字）来分别对 PKU 数据集和 MSR 数据集进行实验。实验结果如表 9 所示。对于池化的方法，采用三种池化的方法，分别为最大值池化（max pooling）、平均值池化（average pooling）、k-max 池化（k-max pooling）。池化操作类似于卷积，不过池化只是用池化核对感受野范围内的元素进行筛选。最大值池化是选择感受野内最大的特征值，平均值池化是对感受野内的特征取平均值，k-max 池化是最大值池化的扩展，用于选取感受野内数值最大的 k 个特征。考虑到网络中模型参数数量对模型拟合程度的影响，在添加了池化层以后，网络相对地会减少一些特征，从而减少模型参数的数量。因此，在池化实验中适当增加了卷积的通道数和隐藏层单元数量以增强模型对学习任务的拟合程度。

表 8 不同模型的 F1 值对比

模型	PKU	MSR
Zhang et al. (2016)	0.9510	0.9700
Liu et al. (2016)	0.9391	0.9521
Cai and Zhao (2016)	0.9550	0.9650
Cai et al. (2017)	0.9580	0.9710
本文方法	0.9669	0.9684

表 9 不同池化方法使用 BE 标签集的实验结果

池化方法	PKU				MSR			
	P	R	F1	A/%	P	R	F1	A/%
Max-pooling	0.8163	0.8005	0.8046	80.05	0.8074	0.7994	0.8015	79.94

Average-pooling	0.8067	0.8057	0.8058	80.57	0.7986	0.7939	0.7952	79.39
4-max-pooling	0.8635	0.8626	0.8628	86.26	0.8685	0.8668	0.8672	86.68
No-pooling	0.9671	0.9669	0.9669	96.69	0.9685	0.9684	0.9684	96.84

实验结果表明，最大值池化和平均值池化对模型性能的表现比较相似，并且比较差。k-max 池化由于是保留了更多的特征，因而表现相对于最大值和平均值池化要好一些，但仍不如无池化层的模型表现。这也验证了前面的假设。

至于移除池化层对模型的时间复杂度影响，本实验测试的结果是有池化层（任意一种池化方式）的模型训练时间是无池化层的模型训练时间的 4 倍左右。也就是说，移除池化层的做法不仅给模型带来准确率的提升，也加快了模型的训练速度。

## 5 结束语

本文在卷积神经网络的基础上，提出一种无池化的卷积神经网络分词模型 PCNN。该模型可以高效地在训练中学习字向量维度之间的特征关联信息，准确地标识出单字的标签类别，从而完成分词任务。该模型准确率高，稳定性好，具有良好的适应性。这是在中文分词领域的一次新的尝试，通过学习字向量特征之间的关系信息来识别字与字之间潜在的“分隔”，从而准确地标识出字的标签类别。并且，实验结果也验证了本文的观点。根据由不同领域的文本数据训练的字向量的不同，该模型也适应各种领域内的中文分词任务。由于实验条件有限，没有进行更多的模型测试，模型还有很大的改进空间，有待于后续的研究。

## 参考文献

- [1] Liu Z, Sun M. Chinese word segmentation with conditional random fields [C]//Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. Barcelona, Spain: Association for Computational Linguistics (ACL), 2004: 488-495.
- [2] Wang Y, Li M, Zhang M, et al. Chinese word segmentation using a novel neural network architecture [J]. Neurocomputing, 2018, 317: 453-461.
- [3] Zhang Y, Clark S. Chinese word segmentation with rich character representations [C]//Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg, PA: Association for Computational Linguistics (ACL), 2014: 284-292.
- [4] Zhang Y, Wang X, Wang S, et al. An end-to-end neural model for Chinese word segmentation [J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2018, 26 (6): 1117-1126.

- [5] Liu Q, Zhang Y, Liu T, et al. Chinese word segmentation with bidirectional long short-term memory networks [J]. arXiv preprint arXiv:1511.04688, 2015.
- [6] Feng Y, Yang M, Li G, et al. A hybrid neural network model for Chinese word segmentation [J]. Applied Sciences, 2019, 9 (14): 2866.
- [7] Li H, Zhang Y, Liu Q, et al. Chinese word segmentation using neural networks with character-level features [C]//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin, Texas: Association for Computational Linguistics (ACL), 2016: 2122-2127.
- [8] Liu X, Li Y, Zhang Y, et al. Chinese word segmentation with a unified neural network architecture [C]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, Denmark: Association for Computational Linguistics (ACL), 2017: 2169-2178.
- [9] Zheng Y, Wang J, Zhang Y. A Chinese word segmentation method based on dictionary and HMM [C]//Proceedings of the 2023 ACM SIGAPP Symposium on Applied Computing. New York, NY, USA: ACM, 2023: 1-8.
- [10] Liu X, Zhang Y, Liu Q, et al. Chinese word segmentation with recurrent neural networks [J]. Journal of Chinese Information Processing, 2017, 31 (1): 1-9.
- [11] Li J, Liu Q, Zhang Y, et al. Chinese word segmentation with convolutional neural networks [J]. Journal of Computer Research and Development, 2017, 54 (7): 1625-1635.
- [12] Wang Z, Zhang Y, Liu Q, et al. Chinese word segmentation with a neural network-based approach [J]. Information Sciences, 2018, 440-441: 488-498.
- [13] Zhou X, Xu G, Zhang M. Chinese word segmentation using a character-based neural network model [J]. IEEE Access, 2019, 7: 137483-137492.
- [14] Yang Y, Zhang Y, Liu Q, et al. Chinese word segmentation with a hybrid neural network model [J]. Knowledge-Based Systems, 2019, 189: 104977.
- [15] Liu X, Zhang Y, Liu Q, et al. Chinese word segmentation with a multi-task learning approach [J]. Information Sciences, 2019, 482: 210-222.