文章编号：1003-0077(2018)02-0029-09

基于双向 LSTM 神经网络模型的中文分词

金 宸[1]，李维华[1]，姬 晨[1]，金绪泽[2]，郭延哺[1]

(1. 云南大学 信息学院，云南 昆明 650503；
2. 河南师范大学 教育学院，河南 新乡 453007)

摘 要：中文分词是中文自然语言处理的基础。分词质量的好坏直接影响之后的自然语言处理任务。目前主流的分词是基于传统的机器学习模型。近年来，随着人工智能大潮的又一次兴起，长短期记忆(LSTM)神经网络模型改进了普通循环神经网络模型无法长期依赖信息的缺点，被广泛应用于自然语言处理的各种任务中，并取得了不错的效果。对中文分词，该文在经典单向 LSTM 模型上进行改进，增加了自后向前的 LSTM 层，设计了双向 LSTM 模型，改进了单向 LSTM 对后文依赖性不足的缺点；并引入了贡献率 α，对前传 LSTM 层和后传 LSTM 层的权重矩阵进行调节，并设计了四个实验，验证了所建模型的正确性和优越性。

关键词：中文分词；自然语言处理；双向 LSTM；贡献率

中图分类号：TP391 **文献标识码**：A

Bi-directional Long Short-term Memory Neural Networks for Chinese Word Segmentation

JIN Chen[1], LI Weihua[1], JI Chen[1], JIN Xuze[2], GUO Yanbu[1]

(1. Science and Engineering Department of YunnanUniversity, Kunming, Yunnan 650503, China;
2. Education Department of Henan Normal University, Xinxiang, Henan 453007, China)

Abstract: Chinese word segmentation(CWS) is a fundamental issue of Chinese language processing(NLP). which affects the subsequent NLP tasks substantially. At present, the state-of-the-art solution is based on the classical machine learning model. Recently, Long Short-term Memory (LSTM) model has been proposed to solve the long-term dependencies in classical RNN model, and already well daapted in various kinds of NLP tasks. As for CWS task, we add a layer of backward LSTM based on unidirectional classical LSTM to build a Bi-directional Long Short-term Memory Neural Network model (Bi-LSTM). And we also propose a contribution rate to balance the matrix's value in forward LSTM layer and backward LSTM layer. We design four experiments to demonstrate that our model is reliable and preferable.

Key words: CWS; NLP; Bi-LSTM; contribution rate

0 引言

中文分词是指将连续的中文字符串按照一定的规范分割成词序列的过程。中文不同于英文，其自身的特点在于中文是以字为基本书写单位，句子和段落之间通过分界符来划界，但词语之间并没有一个形式上的分界符，而在自然语言处理中，词是最小的能够独立运用的有意义的语言成分，所以分词质量的好坏直接影响之后的自然语言处理任务[1]。

中文分词问题作为中文自然语言处理领域的重要基础研究，从 20 世纪 80 年代提出到现在，常用的研究方法可以分为以下四类：(1)基于字典的字符串匹配方法[2-3]；(2)基于语言规则的方法[4-5]；(3)基于传统概率统计机器学习模型的方法；(4)基于深度神经网络模型的方法。

随着 SIGHAN 国际中文分词评测 Bakeoff 的展开，将中文分词任务视为序列标注问题来解决逐

收稿日期：2017-04-03 **定稿日期**：2017-07-06

基金项目：国家自然科学基金(11661081)

渐成为主流。基于传统机器学习模型的方法主要为基于字标注的概率统计机器学习模型方法，在Bakeoff展开的初期，基于字标注的中文分词方法广泛应用，在评测中取得性能领先的系统均应用了此类思想[6]。基于统计的自然语言处理方法在消除歧义和句法分析等方面得到越来越广泛的应用，是近年来兴起的一种新的、也是最常使用的方法。对于给定的输入词串，该方法先确定其所有可能的词性串，选出得分最高的作为最佳输出。其中应用比较广泛的主要有隐马尔可夫模型(hidden markov model，HMM)[7]、最大熵模型(maximum entropy model，MEM)[8]和条件随机场(conditional random fields，CRF)[9-11]。以上基于传统机器学习模型的性能受限于特征的选择和提取，模型的训练是基于提取出的人为设定的特征。

为了尽可能避免特征工程的影响，深度学习网络模型逐渐应用到中文分词等自然语言处理任务中。2011年Collobert[12]将神经网络模型应用到自然语言处理中。2013年，Zheng等人[13]首先将神经网络模型应用到中文分词任务，同时还提出了一种感知器算法，在几乎不损失性能的前提下加速了训练过程。在此基础上，Pei等人[14]通过利用标签嵌入和基于张量的转换，提出了MMTNN的神经网络模型的方法，并用于中文分词任务。2015年，Chen等人[15]使用LSTM神经网络来解决中文分词问题，克服了传统神经网络无法长期依赖信息的问题，取得了很好的分词效果，同年，Chen等人[16]构造了一种基于栈结构的GRU神经网络模型，使用树形结构来捕捉长期依赖信息。这些方法都取得了非常不错的效果。

然而，单向LSTM神经网络只能记住过去的上文信息，但中文句子的结构较为复杂，有时需要联系下文的信息才能做出判断。2015年Huang[17]提出了一种双向LSTM-CRF模型，并把它用在了序列标注的任务上，取得了很好的效果。受此启发，在Chen[15]模型的基础上，本文提出使用双向的LSTM神经网络模型进行分词，在单向LSTM神经网络的基础上增加一层自后向前的LSTM神经网络层，并引入贡献率α对前传LSTM层和后传LSTM层输入隐藏层的权重矩阵进行调节，综合双向的记忆信息，实现更加准确的分词。

1 双向LSTM神经网络模型

1.1 LSTM神经网络模型

RNN(recurrent neural network)模型是Rumelhart等人[18]在1986年提出的具有循环结构的网络结构，具备保持信息的能力。RNN模型中的循环网络模块将信息从网络的上一层传输到下一层，网络模块的隐含层每个时刻的输出都依赖于以往时刻的信息。RNN模型的链式属性表明其与序列标注问题存在着紧密的联系，但在经典RNN模型的训练中，存在梯度爆炸和梯度消失的问题，且经典RNN模型很难处理长期依赖的问题。

LSTM神经网络(Long short-term memory neural network)模型[19]是RNN的扩展，专门设计用来处理长期依赖缺失的问题。与经典RNN网络不同，LSTM的循环单元模块具有不同的结构，存在四个以特殊方式相互影响的神经网络层。

LSTM网络的关键在于LSTM单元的细胞状态。在LSTM单元中，通过门(gates)结构来对细胞状态增加或删除信息，而门结构是选择性让信息通过的方式，如图1所示。LSTM单元具有输入门(input gates)、忘记门(forget gates)和输出门(output gates)三种门结构，用以保持和更新细胞状态，以下公式中i_t、f_t、o_t和C_t表示t时刻对应的三种门结构和细胞状态。

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ \hat{C}_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} W \cdot [h_{t-1}, x_t] \tag{1}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \hat{C}_t \tag{2}$$

$$h_t = o_t \odot \tanh(C_t) \tag{3}$$

LSTM神经网络模型已经在许多应用中取得重大成功，诸如文本、情感分类[20-21]、机器翻译[22]、语意识别[23]、智能问答[24]和对图像进行文本描述[25]等自然语言处理任务中。由于LSTM神经网络模型通过记忆单元去学习从细胞状态中忘记信息、去更新细胞状态的信息，而且具有学习文本序列中远距离依赖的特性，很自然地想到可以使用LSTM神经网络模型进行中文分词的任务。

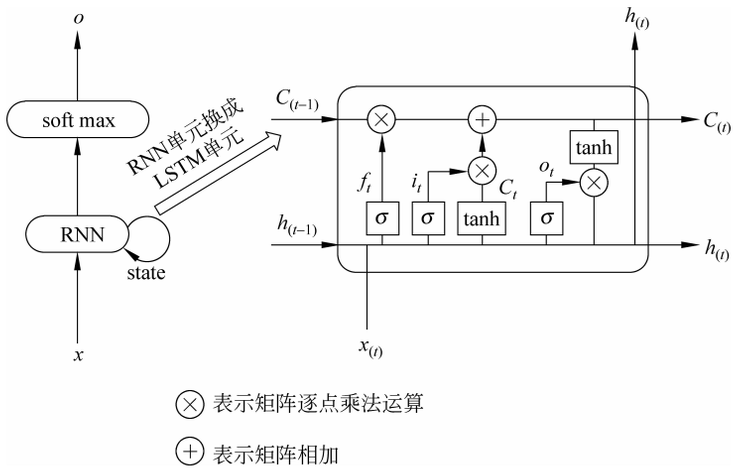

图 1　LSTM 结构图

1.2　双向 LSTM 神经网络模型

双向 RNN(BRNN)模型是 Schuster[26] 在 1997 年提出的，目的是解决单向 RNN 无法处理后文信息的问题，单向的 RNN 只能在一个方向上处理数据，则双向循环神经网络的基本思想是提出每一个训练序列向前和向后分别是两个循环神经网络(RNN)，而且这两个都连接着一个输出层。图 2 展示的是一个沿着时间展开的双向循环神经网络。

其中自前向后循环神经网络层的更新公式为：

$$\overrightarrow{h_t} = H(W_{x\overrightarrow{h}}x_t + W_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h_{t-1}} + b_{\overrightarrow{h}}) \tag{4}$$

自后向前循环神经网络层的更新公式为：

$$\overleftarrow{h_t} = H(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h_{t+1}} + b_{\overleftarrow{h}}) \tag{5}$$

两层循环神经网络层叠加后输入隐藏层：

$$y_t = W_{\overrightarrow{h}y}\overrightarrow{h_t} + W_{\overleftarrow{h}y}\overleftarrow{h_t} + b_y \tag{6}$$

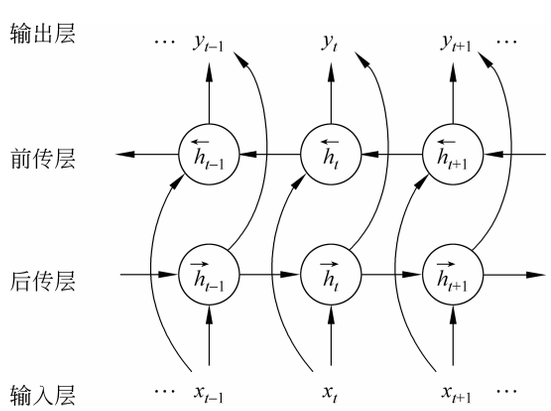

图 2　双向 RNN 结构图

双向 LSTM 神经网络(Bi-direction long short-term memory neural network)模型是结合双向 RNN 和 LSTM 两个模型的优点形成的新模型，简单来说就是用 LSTM 单元替换掉经典双向 RNN 模型中的循环单元。2005 年 Graves[27] 首次将双向 LSTM 神经网络模型应用于分类问题，并取得了较单向 LSTM 神经网络模型更为出色的结果。随后这个模型被推广到自然语言处理的各项任务中：2009 年 Wollmer[28] 将双向 LSTM 模型应用于关键字提取；2013 年 Graves[29] 将其应用于语音识别；2015 年 Wang[30] 将其应用于字嵌入中；2015 年 Huang 将其应用于词性标注[17]；2016 年 Kiperwasser[31] 将其应用于句法分析中。这些应用均取得了很好的效果。

2　基于双向 LSTM 神经网络的中文分词模型

中文分词可视为字符级别的序列标注问题，因此可以将分词过程视为对字符串中每一个字符标注的机器学习过程。目前，学术界使用最广泛的字符标注方法是四词位标注集{B,M,E,S}，其中 B(begin)代表标注词的开始字符，M(middle)代表标注词的中间字符，E(end)代表标注词的结束字符，S(single)代表标注词是单字字符。通过为字符序列中的每一个字符确定相应的标签，我们可将此问题转化为一个多分类的问题，然后通过神经网络模型的多分类层实现相关的标签分类。

基于神经网络的中文分词模型主要由三个部分组成：

(1) 文本向量化层；

(2) 神经网络层；

(3) 标签推断层。

基于双向 LSTM 神经网络的中文分词模型如图 3 所示。

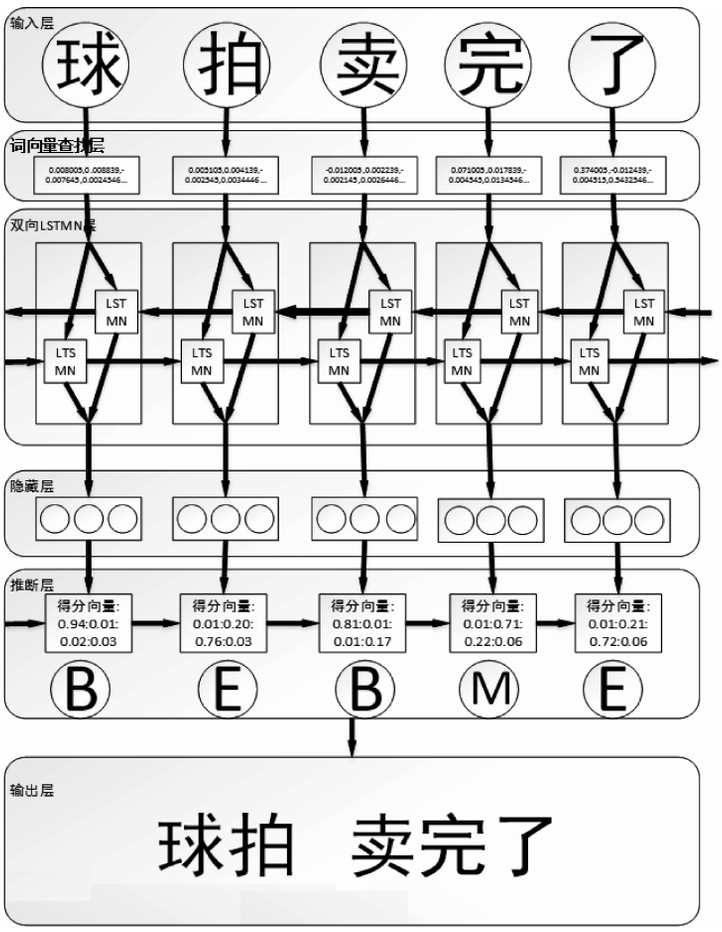

图 3 双向 LSTM 神经网络模型结构图

2.1 文本向量化层

使用神经网络模型来处理数据，需要先将输入的数据进行向量化处理。文本向量化的方式主要有两种。

(1) 独热表示(onehot representation)：就是用一个很长的向量来表示一个词，向量的长度为词典的大小，向量的分量只有一个 1，其它全为 0。1 的位置对应该词在词典中的位置。但这种词表示有两个缺点：

① 会因为词典过大造成数据的维数非常大，而所构成的矩阵非常稀疏，不易进行训练，就是所谓的“维数灾难”问题；

② 不能很好地刻画词与词之间的相似性，也就是所谓的词汇鸿沟问题。

(2) 分布式表示(distributed representation)[32]是针对独热表示这两大缺点而提出的方法[31]。通过训练将某种语言中的每一个词映射成一个固定长度的短向量，将所有这些向量放在一起就形成一个词向量空间，而每一向量则为该空间中的一个点，在这个空间中引入“距离”，就可以根据词之间的距离来判断它们之间的语义相似性了。分布式表示通常又称 embedding 字嵌入(embedding)。

已有的研究表明，加入预先训练的字嵌入向量可以提升自然语言处理任务的性能。Word2Vec[33-34]是 Google 公司于 2013 年开源推出的一个获取字向量的工具包，它简单、高效、易于使用。本文的实验部分用 Word2Vec 作为第一层，把输入数据预先处理成字嵌入向量。基于字标注的分词方法则基于一个局部滑动窗口，假设一个字的标签极大地依赖于其相邻位置的字。给定长度为 n 的文本序列$c^{(1:n)}$，大小为 k 的窗口从文本序列的第一个字$c^{(1)}$滑动至最后一个字$c^{(n)}$。对序列中每个字$c^{(t)}$，当窗口大小为 5 时，上下文信息($c^{(t-2)}$，$c^{(t-1)}$，$c^{(t)}$ $c^{(t+1)}$ $c^{(t+2)}$)将被送入查询表中，当字的范围超过了序列边界时，将以诸如“start”和“end”等特殊标记来补充。然后，将查询表中提取的字向量连接成一个向量$\boldsymbol{X}^{(t)}$。

2.2　双向 LSTM 神经网络层

双向 LSTM 神经网络层由两个部分构成：(1) 自前向后的单层 LSTM；(2) 自后向前的单层 LSTM。

设窗口大小为 k，字向量维度为 d，窗口内的文本数据通过训练好的字嵌入查找表，得到一个分布式表示向量，将此分布式表示向量从前往后输入到一个独立的 LSTM 单元中；又从后往前将其逆序后输入到一个独立的 LSTM 单元中。同时我们引入贡献率变量 α 来调整两个独立的单向 LSTM 层对后续数据的贡献影响，加权之后输入隐藏层进行线性变换，得到一个与标签集维度相等的向量。

2.3　标签得分计算

中文分词问题可以转换成字符序列中字符的标签分类问题。对于字符序列中的每个字符，基于双向 LSTM 神经网络的中文分词模型都会给出一个它在每类标签的得分。

以一个输入序列 $c^{(1:n)}$ 为例，概率 C_t，设窗口大小为 k，字向量维度为 d，则通过训练好的字嵌入查找表，从前往后在 m 时刻得到一个维度为 $k\times d$ 向量 $\boldsymbol{x}_{(mk+1,(m+1)k)}$，输入到一个独立的 LSTM 单元中；从后往前在 m 时刻得到一个维度为 $k\times d$ 向量 $\boldsymbol{x}_{((n-m)k+1,(n-m+1)k)}$，将其逆序后输入到一个独立的 LSTM 单元中。两个输入作为双向 LSTM 神经网络的输入。

通过常识我们判断，对于分词任务来说，自前文的信息量与自后文的信息量是不对等的，前者要大于后者，也就是说通过自前往后 LSTM 层获得的 $g_f(x^{(t)})$ 与通过自后向前 LSTM 层获得的 $g_b(x^{(t)})$ 贡献不同。因此，我们引入一个贡献率变量 α，并且 $\alpha\geqslant 0.5$。在引入 α 的条件下，双向 LSTM 神经网络经过变换之后得到一个输出 $y^{(t)}$，如式(7)所示。

$$y^{(t)}=\alpha g_f(x^{(t)})+(1-\alpha)g_b(x^{(t)}) \tag{7}$$

$y^{(t)}$ 再经过隐藏层的线性变换，可以得到一个与标签集维度相等的向量 $\boldsymbol{y}^{(t)}$，表示 c_t 属于各个标签的得分。

2.4　标签推断层

在{B,M,E,S}标签系统中，相邻标签的分布并不是相互独立的，如标签 B 之后出现标签 B、S 的概率为 0，也就是说标签 B 之后只可能出现标签 M、E。故本文使用 Collobert[12] 提出了标签转移权重矩阵 $\boldsymbol{A}$ 的方法表示这个依赖关系，其中 $\boldsymbol{A}_{ij}$ 表示从标签 i 转移到标签 j 的权重大小。$\boldsymbol{A}_{ij}$ 的值越高，表示标签 i 转移到标签 j 的可能性越大。那么，对于训练数据集中的一个输入字符序列 $c(1:n)$，其标签序列为 $y(1:n)$，则将该字符标签序列的得分定义为 $s(c_{1:n},y_{1:n},\theta)$，如式(8)所示。

$$s(c_{1:n},y_{1:n},\theta)=\sum_{t=1}^{n}(\boldsymbol{A}_{y_{t-1}y_t}+\bar{y}_t) \tag{8}$$

其中，θ 表示模型的各种权重矩阵参数集合，$\bar{y}$ 表示通过神经网络模型的结果矩阵，而 $\bar{y}_t$ 则表示神经网络模型预测 C_t 属于标签 y 的得分，即正确预测的得分。

设输入的句子为 x，该句子正确的标签序列为 y，用 $Y(x)$ 表示所有 x 可能标签序列的集合。定义 $Y(x)$ 中得分最高的预测标签序列为 $\hat{y}$，如式(9)所示。

$$\hat{y}=\underset{y\in Y_x}{\operatorname{argmax}}\quad s(x,y,\theta) \tag{9}$$

其中，$s(x,y,\theta)$ 来自式(8)，是字符标签序列的得分。

2.5　模型训练

我们用 $\Delta(y_i,\hat{y})$ 定义损失函数，如式(10)所示。

$$\Delta(y_i,\hat{y})=\sum_{t}^{n}\eta 1\{y_i^{(t)}\neq\hat{y}^{(t)}\} \tag{10}$$

其中，$1\{y_i^{(t)}\neq\hat{y}^{(t)}\}$ 表示当 $1\{y_i^{(t)}\neq\hat{y}^{(t)}\}$ 为 1 否则为 0，η 是比例调节参数，$\Delta(y_i,\hat{y})$ 则表示了对于输入句 x，标签预测错误数的线性相关值。设训练集为 T，我们引入 l_2 正则化来减小过拟合程度，$\|\theta\|_2$ 是 l_2 范数的正则化项，用来减少参数空间，避免过拟合。λ 用来控制正则化的强度。定义正则化的目标函数 $J(\theta)$ 如式(11)所示。

$$J(\theta)=\frac{1}{T}\sum_{(x,y)\in t}l_i(\theta)+\frac{\lambda}{2}\|\theta\|_2^2 \tag{11}$$

其中，

$$l(\theta)=\max(0,s(x_i,\hat{y}_i,\theta)+\Delta(y_i,\hat{y})-s(x,y,\theta)) \tag{12}$$

训练过程中用 Dropout[35-36] 来控制在模型训练时随机让网络中的某些隐藏层节点不工作，阻止了某些特征仅仅在其他特定特征下才有效果的情况。最后用小批量 AdaGrad 优化算法[37] 对目标函数进行优化，其计算过程中采用误差反向传播[19] 的方式逐层求出目标函数对神经网络各层权值的偏导数，并更新全部权值和偏置值。

3 实验

3.1 实验环境、数据集和评测指标

本文所用实验环境的主要参数为处理器：Intel(R)Core(TM)i7-6700k CPU @ 4.00GHz；图形加速卡：NVIDIA GeForce GTX 1060 6 GB；内存：16GB；操作系统：Ubuntu 16.04 LTS(64bit)；使用Google开源深度学习框架TensorFlow 0.12构建所有神经网络模型进行训练和测试；使用Word2Vec对字向量进行训练预处理。

本文的实验数据集来自当前学术界普遍采用的训练语料和测试语料，其中本文神经网络模型的训练语料和测试语料来自MSRA数据集和PKU数据集，这个由SIGHAN举办的第二届国际中文分词评测Bakeoff 2005所提供的封闭语料。其中训练语料按照通常做法，取90%作为训练集，10%作为开发集，且用来训练词向量的语料混合了搜狗实验室提供的全网新闻数据(SogouCA)以及MSRA数据集和PKU数据集中的训练集。其语料规模如表1所示。

表1 实验所用语料库规模统计信息

数据集	训练集	开发集	测试集
PKU	1 645 048 (5 196)	181 400 (573KB)	172 733 (336KB)
MRSA	3 650 013 (11 302KB)	400 456 (1 240KB)	184 355 (367KB)
SougouCA	2.1GB		

在对中文分词性能的评估中，采用了Bakeoff 2005提供的评分脚本，其中包括分词评测常用的R(召回率)、P(准确率)和F1(召回率和准确率的调和平均值)，以F1值作为评测的主要参考指标。

3.2 实验设计

本文设计了四个实验。

实验一 为了验证文本向量化的必要性，设计了在其他条件都相同的情况下，实验得到通过未使用字嵌入层在PKU数据集中测试数据P、R、F1的值，以及不同维度下的字嵌入层在PKU数据集中测试数据的P、R、F1值，如表2所示。由于独热向量的“维数灾难”问题，故未使用字嵌入层的实验，只使用MSRA数据集和PKU数据集中的训练集和开发集的数据，将其转化为独热表示。而使用字嵌入层的实验则混合使用SogouCA数据集以及MSRA、PKU数据集中训练集和开发集，通过Word2Vec转化为不同维度的词向量。

表2 随着字嵌入维度的变化，分词模型在PKU数据集上评测指标的变化

字嵌入维度	P	R	F1
未预处理	91.9	91.8	91.8
50	95.5	95.1	95.3
100	**96.8**	**96.4**	**96.6**
150	96.1	95.6	95.8

实验二 为了验证Dropout的有效性，并确定合适的丢弃率，设计了不使用Dropout以及Dropout丢弃率为20%和Dropout丢弃率为50%的实验。在保证实验其他参数相同的条件下，测试在MSRA数据集和PKU数据集中每一次迭代后的F1测试数据的变化情况。实验结果如图4所示。

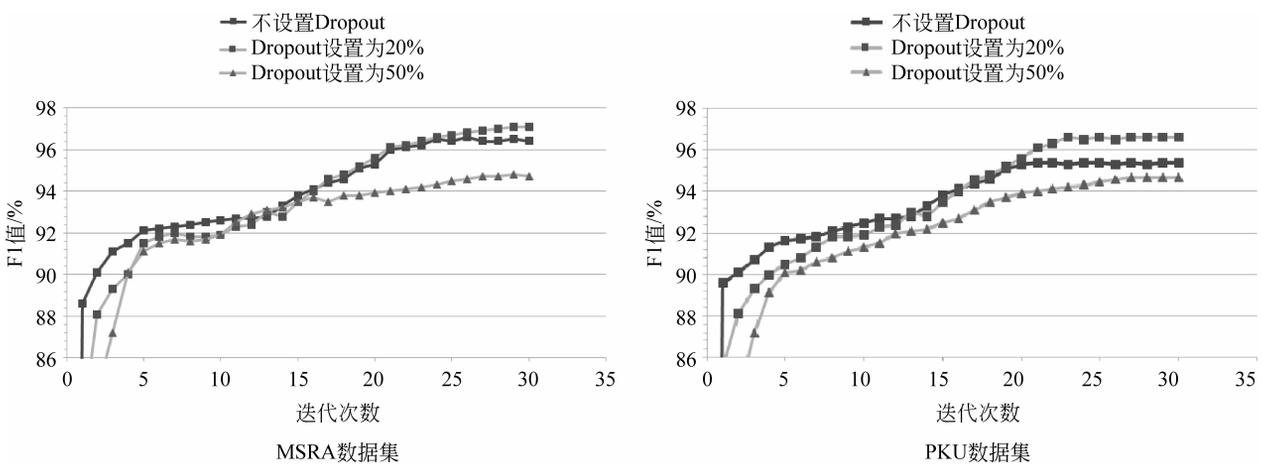

图4 在不同Dropout丢弃率下，本文模型在两个数据集上每次迭代后，分词数据F1的对比

实验三　为了测试本文所构建的双向 LSTM 神经网络模型的效果，本文使用了如下几个基准模型：基于条件随机场模型的分词模型 CRF＋＋[38]；Chen[15]提出的单向 LSTM 分词模型；双向 RNN 分词模型。对基准模型与本文使用的双向 LSTM 分词模型在 MSRA 数据集和 PKU 数据集下进行实验对比，在确保其他变量都一致的情况下（如使用相同维度的字嵌入，在输出层均使用丢弃率相同的 Dropout），记录得到 P、R、F1 测试数据，对比模型参数均基于原作者给出的参数设置，实验统计数据均使用在可信范围内的最佳数据。实验结果如表 3 所示。

表 3　各类分词模型评测指标的对比

经过字嵌入预处理模型	MSRA 数据集			PKU 数据集		
	P	R	F1	P	R	F1
CRF＋＋[38]	92.6	94.0	93.3	93.6	92.1	92.8
LSTM[15]	96.6	96.2	96.4	95.8	95.5	95.7
Bi-RNN	95.7	94.8	95.2	94.2	92.5	93.3
Bi-LSTM（α＝0.8，预处理过词向量）	**97.3**	**97.1**	**97.2**	**96.8**	**96.4**	**96.6**

实验四　为了验证本文提出的贡献率 α 是否会影响到实验效果，并确定效果最佳的贡献率 α，本文设计了六个 α 取值，从 0.5 到 1.00，相邻单位取值相差为 0.1。以六个 α 值为基础构建了本文设计的双向 LSTM 神经网络模型，并保证其他参数都相同的条件下，在 MSRA 数据集和 PKU 数据集下进行分词实验，并得到在不同的贡献率 α 下的测试数据 P、R、F1，并进行对比。实验结果如表 4 所示。

表 4　随着 α 的增长，分词模型评测指标的变化

经过字嵌入预处理模型		MSRA 数据集			PKU 数据集		
α	1－α	P	R	F1	P	R	F1
0.50	0.50	95.6	95.3	95.4	94.3	93.8	94.1
0.60	0.40	95.8	95.4	95.6	94.8	94.3	94.5
0.70	0.30	96.1	95.7	95.9	95.3	95.1	95.2
0.80	**0.20**	**97.3**	**97.1**	**97.2**	**96.8**	**96.4**	**96.6**
0.90	0.10	97.1	96.7	96.9	95.8	96.1	95.9
1.00	0	96.6	96.2	96.4	95.8	95.5	95.7

3.3　实验参数设置

通过多次实验优化参数，我们最终把各项参数设置如下：初始学习率设置为 0.2，最小批处理尺寸设置为 20，隐藏层节点数设置为 150，字嵌入向量的维度为 100。对于输入窗口，我们将窗口分为左右两边，左窗口设置为 0，右窗口设置为 2。即将 t 到 t＋2 的三个字符同时输入。为防止神经网络过拟合，我们采用 l_2 正则化，参数设置成 10^{-4}，同时采用 Dropout，并设置 Dropout 的丢弃率为 0.2。

3.4　实验结果分析

实验一　通过对比表 2 第 2 行和第 3、4、5 行数据可知，文本向量化的处理是非常必要的，加入字嵌入层会极大地提高模型的正确率。由使用大数据集 SougouCA 转化独热表示失败可知：只能在较小的规模下使用独热表示，若训练数据集较大，会导致词典过大而造成数据的维数非常大，且构成的矩阵非常稀疏，不易进行训练。其次，通过对比表 2 第 3、4、5 行数据可知：文本向量化使用的维度也会对结果有一定的影响，故本文采用结果相对较好的 100 维作为字嵌入向量的维度。

实验二　通过观察图 4 中数据点的分布和走向有如下三个方面的结论。（1）不设置 Dropout 的模型在迭代前几次表现得较好，但随着迭代次数的增加，模型评测数据趋于稳定后，Dropout 丢弃率为 20％的模型表现优于不设置 Dropout 的模型；（2）Dropout 丢弃率设置为 50％的模型在整个迭代过程中都表现得比较糟糕，说明 Dropout 的丢弃率不宜过大，过大后可能会丢失重要信息；（3）无论是在 MSRA 数据集还是在 PKU 数据集，二者的趋势都较为接近，说明本文模型在不同数据集上表现较为一致，可以使用相同的参数设置。

实验三　通过对比表 3 第 6 行和第 4、5 行数据可知：本文模型在 MSRA 数据集上实验结果 F1，较单向 LSTM 提升 0.72％，较双向 RNN 提升 1.67％；在 PKU 数据集上的实验结果 F1，较单向 LSTM 提升 1.04％，较双向 RNN 提升 2.76％。通

过数据的分析比较，说明文本所提出的模型在分词的准确度上确有提高。

实验四 通过对表4的各项数据的比较可知：(1)贡献率α对实际分词表现作用比较明显，P、R、F1的值随着α的增长，先变大后变小，在0.8处到达峰值。(2)无论是在MSRA数据集还是PKU数据集，二者的趋势都较为接近，这说明本文模型在不同数据集上表现较为一致，可以使用相同的参数设置。

4 结束语

本文的工作主要有两点：(1)将双向的LSTM神经网络模型运用到中文分词任务中，并构建了完整的模型；(2)创新地引入了贡献率α，通过α对前传LSTM层和后传LSTM层输入隐藏层的权重矩阵进行调节，设计了四个实验，实验结果证明：①使用文本向量化的字嵌入和在输入层设置Dropout会对实验结果带来影响；②本文构建的双向LSTM神经网络中文分词模型在正确率上要优于其他基准模型；③本文提出的贡献率α的确会对实验结果带来影响。

本文模型还存在着以下不足：(1)双向LSTM模型较单向LSTM模型在模型结构上更为复杂，从而在模型训练和测试的时候效率不如单向LSTM模型；(2)由于条件所限，本文实验在设置精度上比较粗糙，并没有优化到最理想的参数设置。

接下来值得研究改进的方向：(1)使用GRU等LSTM的变种单元替换传统LSTM，使得模型进一步简化，在效率上进行提升；(2)引入注意力机制完善模型，争取在正确率上进一步提升；(3)将本文所用的分词模型和贡献率α进一步套用在其他序列标注的相关问题(如词性标注、命名实体识别)上。

参考文献

[1] 黄昌宁，赵海. 中文分词十年回顾[J]. 中文信息学报，2007，21(3)：8-19.

[2] 梁南元. 书面汉语自动分词系统——CDWS[J]. 中文信息学报，1987，1(2)：46-54.

[3] 赵海，揭春雨. 基于有效子串标注的中文分词[J]. 中文信息学报，2007，21(5)：8-13.

[4] Wu A，Jiang Z. Word segmentation in sentence analysis[C]// Proceedings of the 1998 International Conference on Chinese Information Processing，1998：169-180.

[5] Sui Z，Chen Y. The research on the automatic term extraction in the domain of information science and technology[C]//Proceedings of the 5th East Asia Forum of the Terminology，2002.

[6] 任智慧，徐浩煜，封松林，等. 基于LSTM网络的序列标注中文分词法[J]. 计算机应用研究，2017，34(5)：1321-1324.

[7] 李月伦，常宝宝. 基于最大间隔马尔可夫网模型的汉语分词方法[J]. 中文信息学报，2010，24(1)：8-14.

[8] Xue N，Converse S P. Combining classifiers for Chinese word segmentation[C]//Proceedings of the first SIGHAN workshop on Chinese language processing-Volume 18. Association for Computational Linguistics，2002：1-7.

[9] Peng F，Feng F，McCallum A. Chinese segmentation and new word detection using conditional random fields[C]//Proceedings of the 20th International Conference on Computational Linguistics. Association for Computational Linguistics，2004：562.

[10] 罗彦彦，黄德根. 基于CRFs边缘概率的中文分词[J]. 中文信息学报，2009，23(5)：3-8.

[11] 方艳，周国栋. 基于层叠CRF模型的词结构分析[J]. 中文信息学报，2015，29(4)：1-7.

[12] Collobert R，Weston J，Bottou L，et al. Natural language processing(almost)from scratch[J]. Journal of Machine Learning Research，2011，12(1)：2493-2537.

[13] Zheng X，Chen H，Xu T. Deep learning for Chinese-word segmentation and POS tagging[C]//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing，2013：647-657.

[14] Pei W，Ge T，Chang B. Max-margin tensor neural network for Chinese word segmentation[C]//Proceedings of the Meeting of the Association for Computational Linguistics，2014：293-303.

[15] Chen X，Qiu X，Zhu C，et al. Long short-term memory neural networks for Chinese word segmentation[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing，2015：1197-1206.

[16] Chen X，Qiu X，Zhu C，et al. Gated recursive neural network for Chinese word segmentation[C]//Proceedings of the ACL(1)，2015：1744-1753.

[17] Huang Z，Xu W，Yu K. Bidirectional LSTM-CRF models for sequence tagging[J]. arXiv preprint arXiv：1508.01991，2015.

[18] Rumelhart D E，Hinton G E，Williams R J. Learning representations by back-propagating errors[J]. Nature，1986，323(6088)：533-536

[19] Hochreiter S，Schmidhuber J. Long short-term memory[J]. Neural Computation，1997，9(8)：1735-

1780.

[20] Liu P, Qiu X, Chen X, et al. Multi-timescale long short-term memory neural network for modelling sentences and documents[C]//Proceedings of Conference on Empirical Methods in Natural Language Processing: 2326-2335.

[21] Wang X, Liu Y, Sun C, et al. Predicting polarities of tweets by composing word embeddings with Long Short-Term Memory[C]//Proceedings of Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing, 2015: 1343-1353.

[22] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]//Proceedings of the 20th NIPS, 2014: 3104-3112.

[23] Graves A, Mohamed AR, Hinton G. Speech recognition with deep recurrent neural networks[C]//Proceedings of IEEE International Confenence on Acoustics, 2013, 38(2003): 6645-6649.

[24] Wang D, Nyberg E. A long short-term memory model for answer sentence selection in question answering [C]// Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing, 2015: 707-712.

[25] Vinyals O, Toshev A, Bengio S, et al. Show and tell: A neural image caption generator[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015: 3156-3164.

[26] Schuster M, Paliwal K K. Bidirectional recurrent neural networks[J]. IEEE Transactions on Signal Processing, 1997, 45(11): 2673-2681.

[27] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures[J]. Neural Networks, 2005, 18(5): 602-610.

[28] Wollmer M, Eyben F, Keshet J, et al. Robust discriminative keyword spotting for emotionally colored spontaneous speech using bidirectional LSTM networks[C]//Proceedings of the ICASSP 2009. International Conference on IEEE, 2009: 3949-3952.

[29] Graves A, Jaitly N, Mohamed A. Hybrid speech recognition with deep bidirectional LSTM[C]//Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). IEEE, 2013: 273-278.

[30] Wang P, Qian Y, Soong F K, et al. A unified tagging solution: Bidirectional LSTM recurrent neural network with word embedding[J]. arXiv preprint arXiv: 1511.00215, 2015.

[31] Kiperwasser E, Goldberg Y. Simple and accurate dependency parsing using bidirectional LSTM feature representations[J]. arXiv preprint arXiv: 1603.04351, 2016.

[32] Hinton G E. Learning distributed representations of concepts[C]//Proceedings of the eighth annual conference of the cognitive science society. 1986: 1-12.

[33] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[C]// Proceedings of International Conference on Learning Representation, 2013: 1-12.

[34] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[J]. Advances in Neural Information Processing Systems, 2013(26): 3111-3119.

[35] Hinton G E, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing co-adaptation of feature detectors[J]. Computer Science, 2012, 3(4): 212-223.

[36] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

[37] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization[J]. Journal of Machine Learning Research, 2011, 12(7): 2121-2159.

[38] Taku. CRF++: Yet Another CRF toolkit[CP10L]. http://taku910.github.io/crtpp/2005.

金宸(1991—)，硕士研究生，主要研究领域为自然语言处理、机器学习。
E-mail: chenjin0721@gmail.com

姬晨(1993—)，硕士研究生，主要研究领域为自然语言处理。
E-mail: 418445839@qq.com

李维华(1977—)，通信作者，博士，副教授，主要研究领域为数据与知识工程。
E-mail: lywey@163.com