基于词义消歧的卷积神经网络文本分类模型

薛 涛, 王雅玲, 穆 楠
(西安工程大学 计算机科学学院, 西安 710048)

摘 要: 传统文本分类使用 word embedding 作为文档表示,忽略词在当前上下文的含义,潜在地认为相同词在不同文本中含义相同。针对此问题提出一种词义消歧的卷积神经网络文本分类模型——WSDCNN(word sense disambiguation convolutional neural network)。使用双向长短时记忆网络(BLSTM)建模上下文,得到词义消歧后的文档特征图;利用卷积神经网络(CNN)进一步提取对文本分类最重要的特征。在四个数据集上进行对比实验,结果表明,所提出方法在两个数据集,特别是文档级数据集上优于先前最好的方法,在另外两个数据集上得到与此前最好方法相当的结果。

关键词: 文本分类; 卷积神经网络; 长短时记忆网络; 特征提取; 自然语言处理
中图分类号: TP391 **文献标志码**: A **文章编号**: 1001-3695(2018)10-2898-06
doi:10.3969/j. issn. 1001-3695. 2018. 10. 004

Convolutional neural network based on word sense disambiguation for text classification

Xue Tao, Wang Yaling, Mu Nan
(*College of Computer Science, Xi' an Polytechnic University, Xi' an* 710048, *China*)

Abstract: Traditional text categorization usually uses word embedding directly as a representation of the document, ignoring the meaning of the word in the current context, and potentially believes that the same word has the same meaning in different texts. This paper proposed a novel text classification model called WSDCNN(word sense disambiguation convolutional neural network) to solve the above problems. It used the bidirectional long short term memory network (BLSTM) for modeling context, and got the document feature map after finishing word sense disambiguation. Then it used convolutional neural network (CNN) to further extract the local characteristics of the document which was the most important for text categorization. Compared with the state-of-the-art models, the proposed model achieves excellent performance on 2 out of 4 data sets, especially to the document level data set and the other two show the same result as the previous best method.

Key words: text classification; convolutional neural network; long short term memory network; feature extraction; natural language processing(NLP)

0 引言

文本分类作为文本挖掘、自然语言处理、信息检索、问答系统等的重要基础,近年来受到了广泛的关注和快速的发展。传统文本分类方法主要基于机器学习方法,如朴素贝叶斯、K-最近邻和 SVM 算法,其性能取决于人工设计的特征[1];再者,由于人为因素的干扰,往往使用人工进行分类的结果很难达到统一的标准,并且还需要一定的先验知识。

2006 年 Hinton 教授提出深度学习的概念,具有自动从大量无监督数据中学习出任务所需特征的优势。深度学习最先在图像处理和语音识别领域发挥重要的作用,随着深度学习的发展,神经网络方法在自然语言处理(NLP)中表现出显著的实力,卷积神经网络(CNN)能够通过窗口滤波器从局部文本中提取高级特征[2]。长短时记忆(LSTM)网络是一种特殊的循环神经网络(RNN),能够根据全局上下文记忆或忽略特征[3]。

使用深度学习进行文本分类最基本的任务是将文本转换成计算机可以识别的向量表示。传统的文本表示方法使用向量空间模型或是 one-hot 表示。One-hot 表示方法的优点是简单直观,但是这种方法一个明显的缺点就是"词汇鸿沟"现象,即词语是原子性的,无法衡量单词之间的语义相关度。向量空间模型主要存在以下三个问题:a)在 NLP 中主要依赖于分布式假设,即在相同语境中出现的词其语义也相近;b)基于这种方式的向量维度与词典中的词个数呈线性关系,使得特征维度过大而特征又非常稀疏;c)由于词语的数量众多,很容易产生维数灾难的问题,它有效地克服了以上的问题,它就是将高维稀疏的向量映射成低维、稠密的实数向量,本质上是在其维度上编码词的语义特征的特征抽取器,也称为词语的分布式表示或词嵌入(word embedding)[4,5],可以直接计算词语之间的语义相关度。

词义消歧是自然语言处理中一项重要基础的任务,相同的词在不同的语料中含义不同的现象普遍存在。词义消歧任务可以直接影响机器翻译、文本分类等任务的性能。本文将词向量与其上下文进行结合,提出一种词义消歧的神经网络文本分类模型——WSDCNN 模型。a)使用 BLSTM 对上下文进行建模,对比基于窗口的上下文建模方法,可以捕获尽可能远的上下文信息,并且相对于传统的 RNN 方法,可以有效地避免梯度消失或梯度弥散的问题,组合词向量和上下文特征,对当前词进行词义消歧得到文档的特征图,有效避免了同一个词在不同的上下文中使用相同的向量表示,可以认为使用 BLSTM 进行词义消歧;b)在双向 LSTM 之后,使用一个卷积层进一步提取文档的局部重要特征;c)使用最大池化操作自动判断对文本

收稿日期: 2017-05-26; **修回日期**: 2017-07-12

作者简介:薛涛(1973-),男,教授,硕导,博士,主要研究方向为大数据、云计算、自然语言处理(xt73@163. com);王雅玲(1993-),女,硕士研究生,主要研究方向为自然语言处理、数据挖掘;穆楠(1992-),男,硕士研究生,主要研究方向为计算机网络、大数据分析.

分类最重要的特征。结合以上三者,WSDCNN 同时拥有循环神经网络和卷积神经网络的优势,在对词本身进行词义消歧的基础上,既能捕获文本的全局特征,又能捕获文本的局部特征。

1　相关工作

文本分类是自然语言处理中众多任务的基础,如情感分析、关系抽取、问答系统以及垃圾邮件检测等。传统的词袋(bag-of-word)模型和向量空间模型通过统计词频的方法对文本进行建模,将文本看成是无序并且词之间是没有语义关系的[6]。基于深度学习的神经网络模型摆脱了传统人工提取特征的劣势,在文本分类任务中取得了很大的进步。这些模型通常通过一个映射层将高维文本映射到低维稠密的空间中,得到文本的向量表示,然后将这些向量与不同的神经网络结合来解决不同的任务。在文本分类任务中,通常采用循环神经网络和卷积神经网络。

RNN 可以在保持词序信息的基础上隐式地抽取句子表示,并且可以在没有明显句子边界的情况下分析整个文档的语义信息[7]。其最大的特点就是神经元的某些输出可以作为其输入再次传输到神经元,即可以利用先前的信息。尽管 RNN 理论上允许上下文在网络中无限循环,但在实践中,Son 等人[8]表明信息在经典 RNN 结构中迅速变化,并且等同于 8-gram 前馈神经网络。RNN 致命的问题是长期依赖问题,可能会出现梯度消失或梯度弥散的问题,无法有效地利用任意长度的历史信息。并且 RNN 具有语义偏置的问题,因为 RNN 建模时是顺序建模句子或文档,文本中靠后的词占据了更主导的地位;但是并非所有的文档重点都在最后,这可能会影响其生成的语义表示的精确度。LSTM 的出现有效地避免了梯度消失和梯度爆炸的问题,使得 RNN 能够真正解决长时依赖问题。Oualil 等人[9]提出的 LSRC 语言模型在大文本压缩任务上取得了不错的效果,主要思想是使用 RNN 捕获上下文的短时依赖,使用 LSTM 捕获上下文的长时依赖。陈龙等人[10]针对现有缺乏有标注数据的问题,提出一种弱监督深度学习框架,并且使用 LSTM 和 CNN 分别实现了此模型,在情感分析任务中取得了优异的成绩。但标准的 LSTM 只能顺序建模上文信息而忽略下文信息,为了建模目标对象的上下文信息,Graves 等人[11,12]提出了 BLSTM,采用双向 LSTM 捕获文档的双向特征完成因素分类及识别任务。Zhou 等人[13]介绍了带注意机制的 BLSTM 模型,自动抽取对分类任务起决定性作用的特征。Yang 等人[14]提出了一种层次化的注意力网络,分别在单词和句子层面使用注意力机制,在文档分类任务取得了不错的结果。

CNN 通过过滤器可以捕获文本的局部特征,Kim[2]证明在 CNN 模型中简单地使用静态词嵌入和微调超参数就可以在很多 NLP 任务中取得优于传统机器学习方法的性能。Vu 等人[15]在关系分类中使用 CNN 和基本的 RNN,指出 CNN 和 RNN 提供了互补的信息:RNN 考虑句子中所有词的加权组合,而 CNN 为此组合关系抽取最多信息的元组并且只考虑结果的激活。何炎祥等人[16]将基于表情符号的情感空间映射与多通道 CNN 结合,有效增强了捕获情感语义的能力,在微博情感分类得到了很好的分类性能。蔡慧苹等人[17]使用 word embedding 和 CNN 进行情感分析,与传统机器学习方法相比,有了明显的提升。夏从零等人[18]针对传统卷积神经网络忽略长距离依存关系中的句法结构和语义信息的问题,提出一种基于事件卷积特征的文本分类方法,在中文新闻语料多分类任务中取得了有效的成果。Zhou 等人[19]使用 BLSTM 模型结合二维最大池化技术进行文本分类任务,分别在时间维和向量维采用最大池化技术对文本分类任务进行改进。Lai 等人[20]提出了 RCNN 模型,在双向 RNN 之后直接使用最大池化层,在文本分类任务中取得了不错的成绩。但是此模型在使用双向 RNN 后得到词的高级表示之后直接使用最大池化层获得了文档的最主要的特征即全局特征而忽略了文档的局部特征,因为在分类任务中有时局部特征对分类结果的影响比较大。

以上的工作中都忽略了词义消歧对文本分类的影响,潜在地认为相同的词在不同的上下文中含义相同。为了弥补以上工作的缺陷,在 RCNN 的基础上,本文提出了词义消歧的神经网络文本分类模型——WSDCNN 模型。使用 BLSTM 建模上下文的能力,结合词向量本身对当前文档中的词语进行词义消歧,得到词义消歧之后的文档表示;将此文档表示作为 CNN 的输入图谱,利用 CNN 提取文本局部特征的优势进一步提取文档更高级的特征最后再对每个特征图谱进行最大池化操作,捕获文档不同层面的最重要特征完成文本分类任务。这样 WSDCNN 模型同时拥有循环神经网络捕获全局上下文的能力和卷积神经网络捕获文档局部重要特征的优势,在对词本身进行词义消歧的基础上能够对文档进行有效的表示,从而更加适用于文本分类任务。

2　WSDCNN 模型

本文提出的深度模型结构如图 1 所示。该模型主要由四部分组成:双向 LSTM 层、卷积层、池化层和输出层。BLSTM 部分的输入为文档的词向量矩阵,输出为当前位置词与其上下文的组合向量矩阵,得到文档词义消歧之后的初步表示;然后使用卷积层得到文档更高级别的特征;再使用池化层捕获文档最重要的特征;最后使用 Softmax 层对文档进行分类。

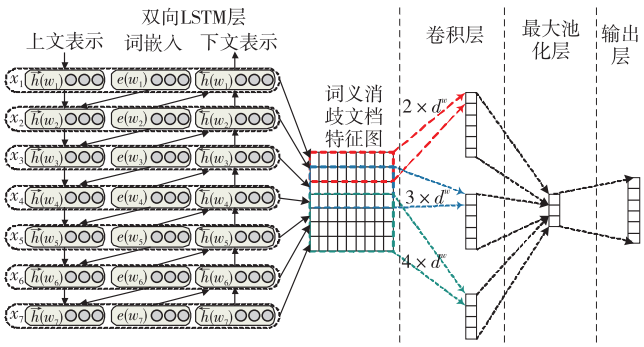

图 1　WSDCNN 模型

算法 1 (Pred, Acc) = WSDCNN(Train, Dev, Test, θ)
输入:训练数据,验证数据,测试数据,网络超参数。
输出:模型预测结果,精确度。

```
Net←buildNetwork()
Net_BLSTM = Net_BLSTM
Net_CNN = Net_CNN
initializeNetwork(Net)
WordEm_train = wordEmMatrix(Train, Dev)
H_train = BLSTM(Net_BLSTM, Train, Dev)
Train_combined = WordEm_train ⊕ H_train
Document_train = CNN(Net_CNN, Train_combined)
Model_softmax = softmax(Document_Train)
error ← error(Model_softmax, Train.labels)
for error > = θ do
    error ←trainNetwork(Net, Train, Dev)
end for
/* traing complete */
WordEm_test = wordEmMap(test)
H_test = BLSTM(Net_BLSTM, test)
Test_combined = WordEm_test ⊕ H_test
Document_test = CNN(Net_CNN, Test_combined)
Pred = softmax(Document_test)
```

```
20  Acc = evaluation(Pred, Test.labels)
21  return(Pred, Acc)
```

模型的伪代码描述如算法 1 所示。其中第 1 ~4 行为模型构建及初始化;第 5 行为使用预训练词嵌入矩阵得到训练和测试数据的词向量,wordEmMatrix 表示预训练词向量;第 6 ~ 13 行为模型训练的过程,使用训练集和测试集数据更新模型参数;第 15 ~20 行为模型测试的过程;第 21 行返回模型预测的结果及精确度。

2.1 词向量

使用深度学习模型进行文本分类时,首先需要将文本转换成计算机可以识别的向量表示,传统的文档表示方法采用词频统计的方法,无法捕获词之间的关系。词向量的出现有效地弥补了这一缺陷,可以将词表示成低维、稠密的实值向量。本文采用词向量作为模型的输入。Turian 等人[5]指出,在大规模无监督语料上学习得到的词向量可以改善模型的效果,为模型提供一个较好的初始值。

本文采用词向量预训练的方式得到文档中的词向量。中英文的词向量使用 word2vec(https://code. google. com/archive/p/word2vec/)对中英文维基百科进行训练。Word2vec 中有 CBOW 和 skip-gram 两种训练词向量的架构,本文使用 skip-gram 架构进行训练。对于文本中的词查询词向量查找表得到对应的词向量作为 BLSTM 的输入,在查找表中不存在的词使用随机初始化的方式进行初始化。

2.2 文档特征图

模型的第一部分使用 BLSTM 对文档上下文进行建模,捕获长时依赖特征。主要可以分为两步,首先使用 BLSTM 进行双向文档建模,得到文档的上下文特征;使用词向量组合上下文特征表示当前词,对文档中的词进行词义消歧得到文档特征图谱,有效避免了传统文本分类方法在不同的文档中使用相同词向量表示词的弊端。

2.2.1 上下文建模

1997 年 Schmidhuber 教授首次提出 LSTM 模型以克服 RNN 的梯度弥散和梯度爆炸的问题[3]。LSTM 的本质属性是记住长期依赖信息,主要思想是引入自适应门控机制,用来决定保持先前隐藏层状态的程度和存储当前数据特征的程度。LSTM 的架构如图 2 所示,主要有输入门 i_t、输出门 o_t 和遗忘门 f_t。凭借着对状态信息的存储和修改,LSTM 单元可以实现长时记忆。

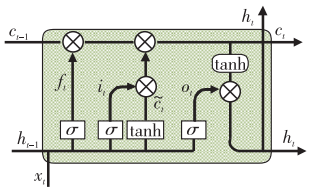

图 2 LSTM 架构

给定文档 $D = \{w_1, w_2, w_3, \cdots, w_l\}$,其中 l 表示文档的长度,w_i 表示文档中的第 i 个词;$\boldsymbol{e}(w_i) \in \mathrm{R}^d$ 是词 w_i 的向量表示,d 是词向量的维度。在 t 时刻,神经元的状态以下面的等式进行更新:

$$f_t = \sigma(W_f \cdot [h_{t-1}, w_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, w_t] + b_i) \tag{2}$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, w_t] + b_c) \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{4}$$

$$o_t = \sigma(W_O \cdot [h_{t-1}, w_t] + b_O) \tag{5}$$

$$h_t = o_t \odot \tanh(c_t) \tag{6}$$

其中:w_t 是当前时刻的输入;i、f、o 分别为输入门、遗忘门和输出门激活;c_t 是当前神经元的状态;o_t 为当前时刻的输出;h_{t-1} 表示当前输入的上文信息;$\odot$ 表示逐点相乘;σ 表示 S 形函数,如 log-Sigmoid 函数、tan-Sigmoid 函数和 ReLU 激活函数等,本文采用 ReLU 激活函数,如式(7)所示。

$$f(x) = \max(0, x) \tag{7}$$

在文本建模中,当前词不仅与上文相关,很多时候与下文也密切相关。本文使用 LSTM 的一个变体 BLSTM 对文档信息进行建模,同时利用文本的上下文信息。本文使用 $\overrightarrow{h}_{t-1}(w_t)$ 表示 t 时刻的上文信息,$\overleftarrow{h}_{t-1}(w_t)$ 表示 t 时刻的下文信息。

2.2.2 词义消歧文档表示

在本文中使用当前词的词向量和其上下文信息的组合表示当前词,目的是为了使用当前词的上下文信息对当前词进行词义消歧。传统模型只使用当前词的上下文信息表示当前词,而忽略了当前词本身的重要性[19];或者使用当前词的预训练词向量直接作为文档表示,潜在地认为词义在不同上下文中不变。本文使用当前词与其上下文的组合能更确定当前词在当前上下文中的含义,起到一定词义消歧的作用。可以看成使用 BLSTM 抽取文档的上下文表示来对当前词进行词义消歧,组合词义消歧之后的词表示得到文档表示。则在 t 时刻词义消歧后的词表示 x_t 为

$$x_t = [\overrightarrow{h}_{t-1}(w_t); \boldsymbol{e}(w_t); \overleftarrow{h}_{t-1}(w_t)] \tag{8}$$

则文档 D 的词义消歧后的文档特征图矩阵表示为

$$H = \{x_1, x_2, \cdots, x_l\} \tag{9}$$

其中:$H \in \mathrm{R}^{l \times d^w}$;$d^w = d + 2d^h$,$d^h$ 是上下文表示的向量维度。

2.3 卷积神经网络

卷积神经网络是目前应用最为广泛的神经网络之一。卷积神经网络与其他神经网络的区别在于其采用权值共享的思想,相邻两层之间只有部分的节点相连,这种结构大大减少了网络中的参数,有效避免了模型过拟合的问题。本文提出的模型是在使用 BLSTM 进行词义消歧之后的文档特征图谱上直接使用 CNN 进行更高级别的抽象特征提取,获得对文本分类相当重要的局部特征。卷积网络一般由卷积层和池化层组成。

2.3.1 卷积层

对于文本分类任务,通常是由文档某些局部特征来决定文档的类别,而不是由整个文档的平均值决定。本文在 BLSTM 后接入一个卷积层,用来捕获词义消歧后文档的局部特征。对特征图矩阵 $\boldsymbol{H}$ 进行卷积操作,过滤器 $m \in \mathrm{R}^{k \times d^w}$ 作用于特征图的 k 个词,且过滤器的大小与词表示的维度相同。过滤器可以将当前层神经网络上的一个子节点矩阵转换为下一层神经网络上的一个单位节点矩阵。则提取出的每一个特征为

$$c_i = f(m \cdot H_{i:i+k-1,1:d^w} + \boldsymbol{b}_c) \tag{10}$$

其中:i 的取值为 $1 \sim l-k+1$;f 是一个非线性激活函数;$\cdot$ 表示点积操作;$\boldsymbol{b}_c \in \mathrm{R}$ 是一个偏置矩阵。将卷积核 m 应用于特征图矩阵 $\boldsymbol{H}$ 得到文档新的特征图:

$$\boldsymbol{C} = \{c_1, c_2, \cdots, c_{l-k+1}\} \tag{11}$$

其中:$\boldsymbol{C} \in \mathrm{R}^{l-k+1}$。经过卷积之后的文档特征图的维度为 $l-k+1$。

上面描述了一个卷积核的情况,在实际的应用中往往使用多个相同维度的卷积核或多个不同维度的卷积核来捕获文档不同层面的高级特征。卷积单元得到的特征值连接起来得到向量 $\boldsymbol{C}$,这个向量只与卷积核的数目有关,不依赖于文本矩阵的长度(文档中单词的个数)。

2.3.2 池化层

卷积层得到的是文档的多个互补的特征图,其维度往往各不相同。为了得到固定维度的文档表示,本文使用最大池化操作来处理卷积之后的结果,得到整个文档的语义表示。

$$\boldsymbol{C}' = \sum_{i=1}^{l-k+1} \max(c_i) \tag{12}$$

池化层的过滤器与卷积层的过滤器类似,本文的最大池化是对卷积层中不同的卷积核得到的文档特征图进行的,因为不同的特征图可以捕捉文档不同方面的特征,将每一个特征图经过最大池化操作得到的文档整体特征连接起来,得到文档的整体特征表示。最大池化得到对文档整体分类起作用的最主要特征,最后得到文档的高级特征抽象表示为

$$A = [C'_1, C'_2, \cdots, C'_n] \tag{13}$$

其中:C'_i 表示第 i 个卷积核得到特征图的最大值;n 为卷积核的个数。使用这种方法可以从不同的角度捕捉到文本的主要特征。池化层可以大大缩小文档的特征矩阵,从而减少最后全连接层中的参数个数,不仅可以加快计算的速度还能有效地防止过拟合问题。

2.4　文档分类

模型的最后一部分是输出层,直接对卷积神经网络的输出特征进行 Softmax 分类。输出层的定义如下:

$$y = WA + b \tag{14}$$

因为式(14)得到的是一个实数值,无法判断文档属于某个类别的概率,所以使用 Softmax 函数对 y 进行归一化,得到文档 D 属于某个类别的概率。则文档 D 属于类别 i 的概率表示如下:

$$P(i|D;\theta) = \frac{\exp(y_i)}{\sum_{k=1}^{m}\exp(y_k)} \tag{15}$$

其中:m 表示类别的个数。则文档预测的分类标签为

$$\hat{y} = \arg\max_i (P(i|D;\theta)) \tag{16}$$

本文中模型训练的目标为最小化分类交叉熵损失:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} t_i \log(p(y_i)) \tag{17}$$

其中:y_i 为文档 D 真实标签的 one-hot 表示;$p(y_i)$ 是 Softmax 分类中每个类别的估计概率;m 为目标类别的个数。

为了防止过拟合问题,本文引入 L_2 正则化。正则化的思想就是在损失函数中加入刻画模型复杂程度的指标。模型的复杂度一般只由权重 W 所决定,因此只对模型中的所有权重使用正则化。最终模型的优化目标为

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} t_i \log(p(y_i)) + \lambda \| w \|_2^2 \tag{18}$$

其中:λ 表示模型复杂损失在总损失中的比例,即正则化项的权重。

3　实验设置

3.1　实验数据集

20News 数据集(http://qwone.com/~jason/20Newsgroups/)是目前应用最广泛的英文文本分类数据集之一,包括 20 个新闻组共约 20 000 封邮件。本文使用"bydate"版本的数据集,在 politics、sports、sciences 和 religion 这四个类别进行实验。该数据集已经将数据集分成了训练集和测试集。

FuDan 数据集(http://www.datatang.com/data/44139; http://www.datatang.com/data/43543)是一个中文文本分类语料,由复旦大学李荣陆教授提供。该数据集已分好训练集和测试集,其中训练集中共包含 9 804 篇文档,测试集中包含 9 833 篇文档,分为 20 个类别。

SST(Stanford sentiment treebank)数据集(https://nlp.stanford.edu/sentiment/)包括 11 855 条电影评论,本文使用五分类(非常正面、正面、中立、负面、非常负面)标签体系,然后去掉中立数据形成二分类标签体系。数据集的统计信息如表 1 所示。

3.2　对比实验

a)基准实验。平均词嵌入是对文本中的所有词取平均,后面直接接上 Softmax 分类器进行分类。此方法直接将文本中的词语映射到低维空间得到词的向量表示,进而得到文档的整体表示,这种方法一般用于神经网络进行分类实验的基础对比实验中。

表 1　文本分类数据集概要信息

数据集	类别数	训练/验证/测试集	平均文档长度	语言
20News	4	5509/612/4074	429	英文
FuDan	20	8823/981/9833	2981	中文
SST-1	5	8544/1101/2210	19	英文
SST-2	2	8550/950/500	19	英文

b)CNN、LSTM、BLSTM。相对于传统的机器学习方法,本文实验采用这三种常用的深度学习方法及其相关的改进模型来证明本文改进方法的性能优势。为了实验的公平性,采用相同的 word embedding 作为模型的输入。

c)RCNN。本文所提出的方法是在 RCNN 模型的基础上改进而来的,所以本文实验用此方法来验证本文方法的有效性。

3.3　超参数设置

本文使用 Stanford tokenizer(https://nlp.stanford.edu/software/tokenizer.shtml)对英文语料进行分词、ICTCLAS(http://ictclas.nlpir.org/)对中文数据进行分词。文本中的停用词和特殊符号均当做普通单词保留。四个数据集均已分好训练集和测试集,其中 SST 已经分好了训练集、验证集和测试集。本文随机选取训练集的 10% 作为验证集,剩下部分作为真实的训练集。根据前人工作,在 20News 数据集上使用 F_1 作为评价指标,其余数据集使用精确度作为评价指标。

网络中超参数的选择一般需要根据数据的不同而有所调节。在实验中,本文参考了 Turian 等人[5]的工作,学习速率 $\alpha = 0.01$,隐藏层大小 $H = 100$,词向量维度 $d = 50$,上下文向量维度 $d^h = 50$。使用 3×150、4×150 和 5×150 的卷积核各 100 个,抽取不同范围的局部高级特征。以往的实验表明模型越复杂、参数越多越能提高分类的性能,但是为了减少模型复杂带来的优化困难问题,本文采用一层卷积层和一层池化层。神经网络的训练方法主要有随机梯度下降算法和批量梯度下降算法,随机梯度下降算法每次只用到一个训练样本就能更新模型的参数,但是容易陷入局部最优;批量梯度下降考虑到全部样本,但是训练中往往耗费大量的存储空间和运算时间,所以本文使用 ADADELTA[21]更新规则通过小批量随机梯度下降算法来训练神经网络,批量设置为 10。权重初始化采用[-1.0, 1.0]的小数进行随机初始化;式(10)中的激活函数 f 使用 ReLU激活函数。

4　实验结果及分析

本文在四个数据集上对比了所提出模型与其他传统模型的分类效果,主要包括本文提出的 BLSTM-Softmax 和 WSDCNN 模型(WSDCNN 模型可以看成是 BLSTM-Softmax 模型的改进)以及传统的 CNN 与 RNN 相关的实验,实验结果如表 2 所示。其中,BLSTM-Softmax 使用双向 LSTM 进行词义消歧之后,再使用 Softmax 进行分类;WSDCNN 是在词义消歧后使用 CNN 提取局部特征,再进行最大池化操作得到文本整体特征来进行分类。

4.1　模型整体性能

从表 2 的实验结果可以看出,使用卷积神经网络和循环神经网络及其混合模型都比传统的基于分类器的方法有明显的优势。本文所提出的 BLSTM-Softmax 和 WSDCNN 模型不仅适用于英文文本分类,同时也可用于中文语料。在四个数据集中,本文所提出 WSDCNN 模型在 20News 和 FuDan 两个数据集中的精确度分别为 96.85% 和 96.70%,都达到了最优的结果;

在其他两个数据集上也都达到了近似最优的结果，在 SST-1 和 SST-2 数据集上分别得到了 52.3% 和 89.32% 的精确度，足以证明本文提出的模型在这四个数据集上的有效性。以往的大部分技术都是在句子、段落以及含有少量句子的短文本中进行实验，为了验证本文提出方法的通用性，在 20News 数据集上进行了实验，通过结果可以看出，本文提出的方法同样适用于长文本分类任务，并且在长文本上取得了最优的结果。

表 2 本文模型与其他模型精确度对比实验结果 /%

模型	SST-1	SST-2	20News	FuDan
average embedding	32.70	79.74	89.39	86.89
CNN[2]	47.15	87.15	–	94.04
DCNN[22]	48.43	86.35	–	–
RCNN[20]	47.21	87.27	96.49	95.20
LSTM[23]	46.83	84.95	93.54	92.48
BLSTM[23]	47.70	87.32	94.0	93.0
BLSTM-2DPooling[19]	49.93	88.15	95.34	94.8
BLSTM-2DCNN[19]	**52.38**	**89.40**	96.32	–
BLSTM-Softmax	51.37	89.10	95.80	96.63
WSDCNN	52.30	89.32	**96.85**	**96.70**

图 3 展示了本文所提出来的两个模型在四个数据集上相对于此前最好的结果其错误率的降低比率。在 FuDan 数据集上，本文提出的模型取得了最好的效果，比此前最好的结果分类误差分别减少了 29.8%、31%。WSDCNN 模型在 20News 和 FuDan 数据集上均得到了最好的结果，相对于此前最好的结果其分类误差分别减少了约 10%、31%。20News 和 FuDan 数据集都是长文本数据，因此可以认为是由于实验数据集的长度所影响的。从实验结果可以推出本文提出的模型更适用于长文本分类，对于短文本也能提高其分类精度，只是效果不是很明显。

4.2 词义消歧对模型的影响

为了验证词义消歧对分类任务的影响，本文在四个数据集上对比了传统 BLSTM 方法与本文提出的增加词义消歧的 BLSTM-Softmax 方法。

图 4 展示了本文提出的通过 BLSTM 进行词义消歧之后使用 Softmax 进行分类的模型与传统的 BLSTM 文本分类模型的精确度对比图。在四个数据集上 BLSTM-Softmax 都表现出了比传统 BLSTM 更好的效果，因为这两个模型唯一的区别就在于词表示的不同，本文的模型增加了词义消歧的词表示，可以证明是由于本文结合了词向量和其上下文表示词，对文档中的词进行词义消歧的结果。因此可以反映本文所提出的词义消歧的文档表示的有效性。

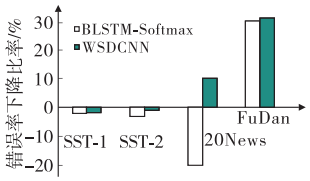

图 3 本文模型相对于此前最好结果错误率降低情况

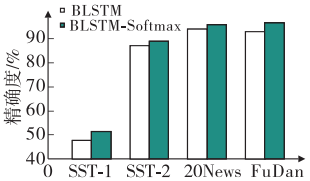

图 4 BLSTM-Softmax 与传统 BLSTM 对比

4.3 WSDCNN 混合模型的优势

图 5 展示了本文提出的 WSDCNN 模型与单独使用 CNN、LSTM 及 BLSTM 模型效果的对比图，可以看出本文混合使用 BLSTM 和 CNN 比单独使用深度学习模型的效果好。从实验结果中可以得出在四个数据集上本文的 WSDCNN 模型精确度分别达到了 52.3%、89.32%、96.85% 和 96.7%，比单独使用 LSTM 分类错误率分别降低了 9.7%、16.9%、–、44.6%，比单独使用 CNN 分类错误率分别降低了 10.2%、29%、51.2%、56.1%。由此可见对于本文实验的数据集使用 BLSTM 进行词义消歧之后与 CNN 结合的方法分类精度更高，通过充分利用 BLSTM 捕获文本长时依赖的上下文特征以及 CNN 捕获文本局部重要特征的优势，进一步提高了文本分类任务的精确度。

4.4 卷积核大小对 WSDCNN 模型影响

在超参数的设置部分，本文选取了 3×150、4×150 和 5×150 的卷积核各 100 个，抽取不同范围的局部高级特征。为了得到选取什么尺寸的卷积核可以得到更精确的分类结果，本文在 SST-1 数据集上进行实验，分别使用 3×150、4×150 和 5×150 卷积核各 300 个进行对比实验。池化层选用最大池化对每个卷积核得到的特征图进行池化，得到文档固定的维度 300。本文在 SST-1 数据集上重复进行五次实验，取其平均值作为最终结果。实验结果如图 6 所示。

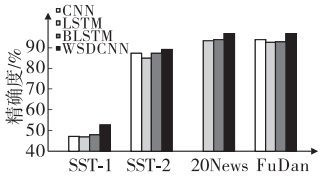

图 5 WSDCNN 与单独的神经网络模型对比

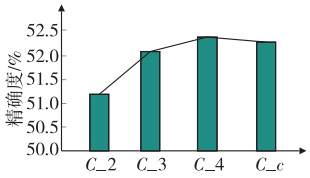

图 6 不同卷积核对精确度的影响

图 6 中，$C\_i(i=2,3,4)$ 表示选用的卷积核尺寸，即滑动窗口的大小；C_c 表示窗口 2、3、4 的组合。可以看出不同的卷积核尺寸可以得到不同的性能，随着尺寸的增大，CNN 可以捕获更多的特征，但是其需要的存储空间和计算时间也会大大增加。在选用卷积核为 4 时，卷积层的参数个数为 4×150×300 = 180 000 个，选用卷积核尺寸为 2、3、4 各 100 个时，卷积层的参数个数为(2+3+4)×150×300 = 135 000，相较于卷积核为 4，其参数个数减少了 25%，而分类精确度相当。因此本文为了模型训练简单起见，选用不同的卷积核尺寸，在兼顾到计算简单的情况下能得到相对好的结果。

5 结束语

本文介绍了两种组合模式，在使用双向 LSTM 进行文本特征提取后，使用词向量和词的上下文表示结合来对文本中的词重新表示，以达到词义消歧的目的。在此基础上 BLSTM-Softmax 是直接使用 Softmax 对词义消歧之后的文档表示进行分类，WSDCNN 是使用 CNN 对词义消歧之后的特征图谱进行局部特征提取来得到文档的更高级抽象表示。本文在 SST-1 数据集上进行深入分析还发现，卷积核尺寸越大其分类精度越高。在四个文本分类任务上的实验结果证明了本文提出模型的有效性。然而该项工作并没有研究此模型在其他 NLP 任务中的应用可能，这将是未来研究的方向。

参考文献：

[1] Li Juntao, Cao Yimin, Wang Yadi, *et al*. Online learning algorithms for double-weighted least squares twin bounded support vector machines [J]. Neural Processing Letters, 2017, 45(1): 319-339.

[2] Kim Y. Convolutional neural networks for sentence classification [C]//Proc of Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA: Association for Computational Linguistics, 2014: 1746-1751.

[3] Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural Computation, 1997, 9(8): 1735-1780.

[4] Mandelbaum A, Shalev A. Word embeddings and their use in sentence classification tasks [J/OL]. (2016-10-26). https://arxiv.org/pdf/1610.08229.pdf.

[5] Turian J, Ratinov L, Bengio Y. Word representations: a simple and general method for semi-supervised learning [C]//Proc of the 48th

Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: Association for Computational Linguistics, 2010: 384-394.

[6] Pang Bo, Lee L, Vaithyanathan S. Thumbs up? Sentiment classification using machine learning techniques [C]//Proc of ACL Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA: Association for Computational Linguistics, 2002: 79-86.

[7] Mikolov T, Karafiát M, Burget L, *et al*. Recurrent neural network based language model [C]//Proc of the 11th Annual Conference of the International Speech Communication Association. 2010: 1045-1048.

[8] Son L H, Allauzen A, Yvon F. Measuring the influence of long range dependencies with neural network language models [C]//Proc of the NAACL-HLT Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT. Stroudsburg, PA: Association for Computational Linguistics, 2012: 1-10.

[9] Oualil Y, Singh M, Greenberg C, *et al*. Long-short range context neural networks for language modeling [C]//Proc of Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA: Association for Computational Linguistics, 2016: 1473-1481.

[10] 陈龙, 管子玉, 何金红,等. 情感分析研究进展[J]. 计算机研究与发展, 2017,54(6):1150-1170.

[11] Graves A, Fernández S, Schmidhuber J. Bidirectional LSTM networks for improved phoneme classification and recognition [C]//Proc of International Conference on Artificial Neural Networks. Berlin: Springer-Verlag, 2005: 753-753.

[12] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures [J]. Neural Networks, 2005, 18(5/6): 602-610.

[13] Zhou Peng, Shi Wei, Tian Jun, *et al*. Attention-based bidirectional long short-term memory networks for relation classification [C]//Proc of the 54th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: Association for Computational Linguistics, 2016: 207-212.

[14] Yang Zichao, Yang Diyi, Dyer C, *et al*. Hierarchical attention networks for document classification [C]//Proc of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016: 1480-1489.

[15] Vu N T, Adel H, Gupta P, *et al*. Combining recurrent and convolutional neural networks for relation classification [C]//Proc of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016:534-539.

[16] 何炎祥, 孙松涛, 牛菲菲,等. 用于微博情感分析的一种情感语义增强的深度学习模型[J]. 计算机学报, 2017, 40(4): 773-790.

[17] 蔡慧苹, 王丽丹, 段书凯. 基于 word embedding 和 CNN 的情感分类模型[J]. 计算机应用研究, 2016, 33(10): 2902-2905, 2909.

[18] 夏从零, 钱涛, 姬东鸿. 基于事件卷积特征的新闻文本分类[J]. 计算机应用研究, 2017, 34(4): 991-994.

[19] Zhou Peng, Qi Zhenyu, Zheng Suncong, *et al*. Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling [C]//Proc of the 26th International Conference on Computational Liguistics. 2016: 3485-3495.

[20] Lai Siwei, Xu Liheng, Liu Kang, *et al*. Recurrent convolutional neural networks for text classification [C]//Proc of National Conference of the American Association for Artificial Intelligence. Palo Alto, CA: AAAI Press, 2015: 2267-2273.

[21] Zeiler M D. ADADELTA: an adaptive learning rate method [J/OL]. (2012-12-22). http://arxiv.org/abs/1212.5701.

[22] Blunsom P, Grefenstette E, Kalchbrenner N. A convolutional neural network for modelling sentences [C]//Proc of the 52nd Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: Association for Computational Linguistics, 2014:655-665.

[23] Tai K S, Socher R, Manning C D. Improved semantic representations from tree-structured long short-term memory networks [C]//Proc of the 53rd Annual Meeting of the Association for Computational Linguistics & the 7th International Joint Conference on Natural Languages Processing. Stroudsburg, PA: Association for Computational Linguistics, 2015: 1556-1566.

(上接第 2897 页)

[14] Kadetotad D, Arunachalam S, Chakrabarti C, *et al*. Efficient memory compression in deep neural networks using coarse-grain sparsification for speech applications [C]//Proc of the 35th International Conference on Computer-Aided Design. New York: ACM Press, 2016: Article No 78.

[15] Vanhoucke V, Senior A, Mao M Z. Improving the speed of neural networks on CPUs [C]//Proc of Deep Learning & Unsupervised Feature Learning Workshop. 2011:1-8.

[16] Hwang K, Sung W. Fixed-point feedforward deep neural network design using weights +1, 0, and −1[C]//Proc of IEEE Workshop on Signal Processing Systems. Piscataway, NJ: IEEE Press, 2014: 1-6.

[17] Chen Wenlin, Wilson J T, Tyree S, *et al*. Compressing neural networks with the hashing trick [C]//Proc of the 32nd International Conference on Machine Learning. 2015: 2285-2294.

[18] Gong Yunchao, Liu Liu, Yang Ming, *et al*. Compressing deep convolutional networks using vector quantization [J/OL]. (2014-12-18). https://arxiv.org/abs/1412.6115.

[19] Hanson S J, Pratt L Y. Comparing biases for minimal network construction with back-propagation [C]//Proc of the 1st International Conference on Neural Information Processing Systems. Cambridge, MA: MIT Press, 1989: 177-185.

[20] LeCun Y, Denker J S, Solla S A. Optimal brain damage [C]//Advances in Neural Information Processing Systems. San Francisco: Morgan Kaufmann Publishers Inc, 1990: 598-605.

[21] Hassibi B, Stork D G. Second order derivatives for network pruning: optimal brain surgeon [C]//Advances in Neural Information Processing Systems. San Francisco: Morgan Kaufmann Publishers Inc, 1992: 164-171.

[22] Han Song, Pool J, Tran J, *et al*. Learning both weights and connections for efficient neural networks [C]//Proc of the 28th International Conference on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2015: 1135-1143.

[23] Han Song, Mao Huizi, Dally W J. Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding [J/OL]. (2016-02-15). https://arxiv.org/abs/1510.00149.

[24] LéCun Y, Bottou L, Bengio Y, *et al*. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[25] Sainath T N, Kingsbury B, Sindhwani V, *et al*. Low-rank matrix factorization for deep neural network training with high-dimensional output targets [C]//Proc of IEEE International Conference on Acoustics, Speech and Signal Processing. Picataway, NJ: IEEE Press, 2013: 6655-6659.

[26] Denil M, Shakibi B, Dinh L, *et al*. Predicting parameters in deep learning [J/OL]. (2014-10-27). https://arxiv.org/abs/1306.0543.

[27] Nakkiran P, Alvarez R, Prabhavalkar R, *et al*. Compressing deep neural networks using a rank-constrained topology [C]//Proc of the 16th Annual Conference of the International Speech Communication Association. 2015: 1473-1477.

[28] Denton E, Zaremba W, Bruna J, *et al*. Exploiting linear structure within convolutional networks for efficient evaluation [C]//Proc of the 27th International Conference on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2014: 1269-1277.