

# 基于 CNN 和双向 LSTM 融合的实体关系抽取

张晓斌, 陈福才, 黄瑞阳

(国家数字交换系统工程技术研究中心, 河南 郑州 450002)

**摘要:** 实体关系抽取旨在识别网络文本中的实体, 并提取出文本中实体之间隐含的关系。研究表明, 深度神经网络在实体关系抽取任务上具有可行性, 并优于传统关系抽取方法。目前的关系抽取方法大都使用卷积神经网络 (CNN) 和长短期记忆神经网络 (LSTM), 然而 CNN 只考虑连续词之间的相关性而忽略了非连续词之间的相关性。另外, LSTM 虽然考虑了长距离词的相关性, 但提取特征不够充分。针对这些问题, 提出了一种 CNN 和 LSTM 结合的实体关系抽取方法, 采用 3 种结合方法进行了实验, 验证了该方法的有效性, 在  $F1$  值上有一定的提升。

**关键词:** 实体关系抽取; 卷积神经网络; 长短期记忆网络; 注意力机制

**中图分类号:** TP393

**文献标识码:** A

**doi:** 10.11959/j.issn.2096-109x.2018074

## Relation extraction based on CNN and Bi-LSTM

ZHANG Xiaobin, CHEN Fucui, HUANG Ruiyang

China National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China

**Abstract:** Relation extraction aims to identify the entities in the Web text and extract the implicit relationships between entities in the text. Studies have shown that deep neural networks are feasible for relation extraction tasks and are superior to traditional methods. Most of the current relation extraction methods apply convolutional neural network (CNN) and long short-term memory neural network (LSTM) methods. However, CNN just considers the correlation between consecutive words and ignores the correlation between discontinuous words. On the other side, although LSTM takes correlation between long-distance words into account, the extraction features are not sufficiently extracted. In order to solve these problems, a relation extraction method that combining CNN and LSTM was proposed. three methods were used to carry out the experiments, and confirmed the effectiveness of these methods, which had some improvement in  $F1$  score.

**Key words:** relation extraction, convolution neural networks, long short-term memory, attention mechanism

## 1 引言

在互联网迅速普及的今天, 数据呈现爆炸性增长的态势, 如何在海量数据中准确高效地获取用户所需要的信息, 成为当前研究的热点。其中,

信息抽取是一个知识发现的过程, 它将网络中的非结构化或半结构文本甚至媒体信息转换为可被机器直接应用的有规则信息。目前的信息抽取技术, 主要分为实体抽取、关系抽取和事件抽取, 本文主要研究关系抽取。关系抽取解决

收稿日期: 2018-06-10; 修回日期: 2018-08-20

通信作者: 张晓斌, zhangxiaobin@pku.edu.cn

基金项目: 国家自然科学基金创新群体基金资助项目 (No.61521003); 国家重点研发计划基金资助项目 (No.2016YFB0800101)

**Foundation Items:** The National Natural Science Fund for Creative Research Groups Project (No.61521003), The National Key Research and Development Program of China (No.2016YFB0800101)

了文本中实体之间关系的分类问题,也是构造复杂知识库系统和知识图谱的关键基础步骤,作为一个底层要素,关系抽取已被广泛应用于诸如智能检索、问答系统、机器翻译、文本摘要等智能应用。

关系抽取任务可以定义为:给定一段文本 $T$ ,以及该文本中包含的实体对 $\langle e_1 \rangle$ 和 $\langle e_2 \rangle$ ,确定文本中该实体对中实体之间的关系,举例如下。

The school  $\langle e_1 \rangle$  master  $\langle e_2 \rangle$  teaches the lesson with a  $\langle e_2 \rangle$  stick  $\langle e_2 \rangle$ .

句子中被标签 $\langle e_1 \rangle$ 和 $\langle e_2 \rangle$ 包围的单词 master 和 stick 就是句子中的 2 个实体,它们在原始数据中已被标记。关系抽取的目的就是提取出这 2 个实体之间的关系,该关系在 SemEval-2010 task 8 数据集<sup>[1]</sup>中表示为 Instrument-Agency,即表示一种工具的使用关系。

现有的关系抽取算法,主要分为基于规则的方法<sup>[2]</sup>、基于核函数的方法和基于机器学习的方法。基于规则的方法需要提取大量的规则,然后用这些规则模板对文本进行匹配,该方法对规则提取要求较高,需要相关人员具有大量的语法知识、词汇特征知识和一定的文学储备。同时,制定规则耗时又耗力,很难进行大规模多领域的信息抽取。基于核函数的方法保留了语料本身的语法信息,能够较充分挖掘文本中的依赖或语法信息。其通过构造核函数计算 2 个具有一定结构(如语法树结构)的对象的相似性。在英文数据集上的关系抽取实验中,已有多种核函数被使用<sup>[3]</sup>,如定义在浅层语法树和依赖树上的层次核、最短路径依赖核和卷积语法树核,以及复合核等。该方法有效减少了规则和特征的提取,但多样的语义表达方式可能会产生不同的结构,不同结构中可能存在着人们无法识别的噪声,这些都会影响核函数方法的性能。基于机器学习的方法则需要启发式地提取可用于关系抽取的特征集合(如词汇、句法、语义等特征),并构建向量,让机器自主地学习,从而实现关系分类。同时,随着深度学习的发展,越来越多的研究倾向于使用神经网络解决问题,深度神经网络可以通过学习自动提取特征,从而减少人工特征的整理,同时也会发现更多隐含的难以发现的特征。但该方法需要大

量人工标注的训练语料,人工语料的构建也是耗时耗力。

目前的神经网络方法,主流是基于卷积神经网络和循环神经网络的方法,但它们往往只使用单一模型,没有充分利用各自模型的优点。基于 CNN 的模型由于卷积核的多样性能够提取较充分的特征;而基于 RNN 的模型则能够充分考虑长距离词之间的依赖性,保留了词序等特征。为了考虑长词依赖和词序等特征的同时,提取更多特征,本文将 CNN 模型和 RNN 模型融合,通过在公开数据集 SemEval-2010 task 8 的实验与现有方法进行对比,证明了模型的有效性。

## 2 相关工作

目前两大主流的神经网络,即卷积神经网络(CNN, convolution neural networks)和循环神经网络,已广泛应用于人工智能领域的各种场景,如图像识别、目标检测、视频追踪、机器翻译、文本分类、情感分析等。在自然语言处理领域,由于词向量的引入,使神经网络的应用成为可能,神经网络方法不同于早期基于机器学习的方法,其无需手动创建大量特征。近年来,已有许多学者将神经网络方法应用于关系抽取,并在各类关系抽取任务中取得优于传统方法的结果。

### 2.1 卷积神经网络

卷积神经网络是深度学习技术中极具代表的网络结构之一(网络结构如图 1 所示),其丰富的卷积核可以用于提取各类特征,虽然这些特征大多都是不可解释的,但却是有效的,因此 CNN 能够自动发现一些文本中隐含的隐性特征。较早将 CNN 用于关系抽取的是 Liu 等<sup>[4]</sup>,加入少量人工特征(如词性标注和实体类别等),将其映射到低维空间后联合词向量作为 CNN 的输入。之后,Zeng 等<sup>[5]</sup>在其基础上增加了位置向量,该向量用于表征实体在句子中的位置信息,实验表明位置向量对最终 F1 的结果增加 9%左右,以至于后来许多学者都沿用了这个特征。Zeng 等<sup>[6]</sup>基于实体位置这个关键信息,提出了分段池化的思想,即在池化层进行池化时,根据实体的位置,将待池化的向量分为 3 段,再分别对这 3 段取最大值进行拼接。

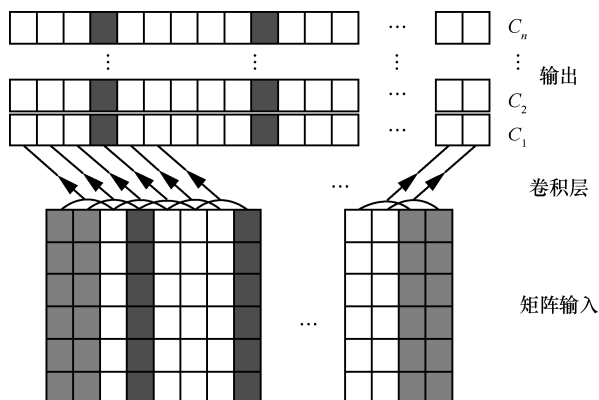


图 1 CNN 网络结构

## 2.2 循环神经网络

卷积神经网络虽然具有较好的学习和特征提取能力，但只能提取短距离内词之间的依赖特征，对于长距离词与词之间的依赖性，却不能较好地提取。基于这种考虑，Zhang 等<sup>[7]</sup>将循环神经网络应用于关系抽取中，解决了长距离词之间的依赖问题。Zhang 等<sup>[8]</sup>使用了双向 LSTM 模型进行关系抽取，取得了不错的效果，证实了双向 LSTM 在关系抽取上的有效性。Zhou 等<sup>[9]</sup>将双向 LSTM 中每一时刻的输出连接起来，引入注意力机制，着重考虑词对关系分类的影响程度。

## 3 模型介绍

图 2 给出了本文提出的神经网络模型的基本结构。首先对数据进行预处理，得到可以作为模型输入的数据；在向量层进行空间映射，得到向量矩阵；之后引入注意力机制，提高模型对个别词汇的关注。接下来，模型采用 3 种方法将 CNN 和 LSTM 融合，融合方法下面具体介绍。经过模型的处理后，将其应用到关系分类，得到最终的结果。

### 3.1 数据预处理与向量映射

本文所使用的 SemEval-2010 tasks 8 数据集已标明了实体所在位置，以及最后的关系类别。对原始数据的预处理，按如下步骤进行。

1) 清洗工作，如大写转换为小写，标点符号的分离等。

2) 使用 NLTK 的 pos\_tag 工具对数据进行词性标注。词性标注是对词的更粗粒度的表示，

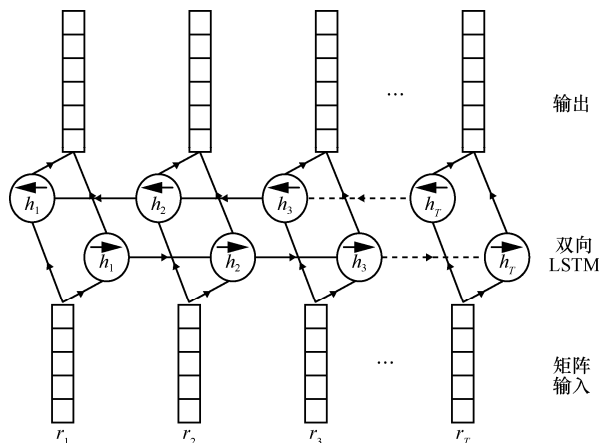


图 2 双向 LSTM 网络结构

如名词表示为 NN，介词表示为 IN，形容词 JJ 等，NLTK 的 pos\_tag 工具使用了 PTB（Penn TreeBank）项目的词性标注方案，包含 36 个词性标签。

3) 使用 Word2Vec<sup>[10]</sup>对单词进行训练。这里先使用 Word2Vec 在大量维基语料上训练，完成后，保留模型的权重参数，紧接着在 SemEval-2010 task 8 数据集上进行训练，可以在一定程度上提高词向量的表示性能。

4) 对于词性标注，由于其类别数量较少，故采用随机初始化来初始化向量。

5) 对于位置向量，还要对句子的每个单词进行标记，对实体位置标记为 0，实体左边的依次标记为 -1, -2, -3 ... 右边的标记为 1, 2, 3 ... 之后再对应到随机初始化的向量，因为有 2 个实体，所以要对对应到 2 个向量。

6) 通过数据预处理和向量映射后，原始数据的句子就可以用（Word, POS-tag, P1, P2）表示，分别为词向量、词性标注向量、实体 1 位置向量、实体 2 位置向量。

### 3.2 双向 LSTM 层

在自然语言处理中，文本的一个显著特点就是序列化，单词出现的顺序往往跟这句话的语义有很大联系。为了充分利用句子结构的信息，本文采用双向 LSTM 模型，考虑正向和反向语序的影响，双向 LSTM 之所以能够提取句子的正反序语义信息，是由双向 LSTM 的网络结构决定的，它由正向 LSTM 和反向 LSTM 组成，两者结构相同，相互独立，只是接受不同的语序输入，因此，最终得

到的隐层向量包含该句子正反语序的信息，更好地提取了句子结构的语义信息。

LSTM 网络是一种特殊的循环神经网络。最早由 Hochreiter 和 Schmidhuber<sup>[11]</sup>于 1997 年提出，解决了传统 RNN 网络的梯度消失问题，以及该问题导致的长时间依赖问题，即不能整合过长序列的相关依赖信息。一个 LSTM 由一个记忆单元  $c$  和 3 个门（输入门  $i$ 、输出门  $o$  和遗忘门  $f$ ）构成，其网络架构如图 3 所示。通过这些结构，在一个时间节点  $t$ ，LSTM 可以选择记住和遗忘某些信息，并将它们传递给下一时刻  $t+1$ 。

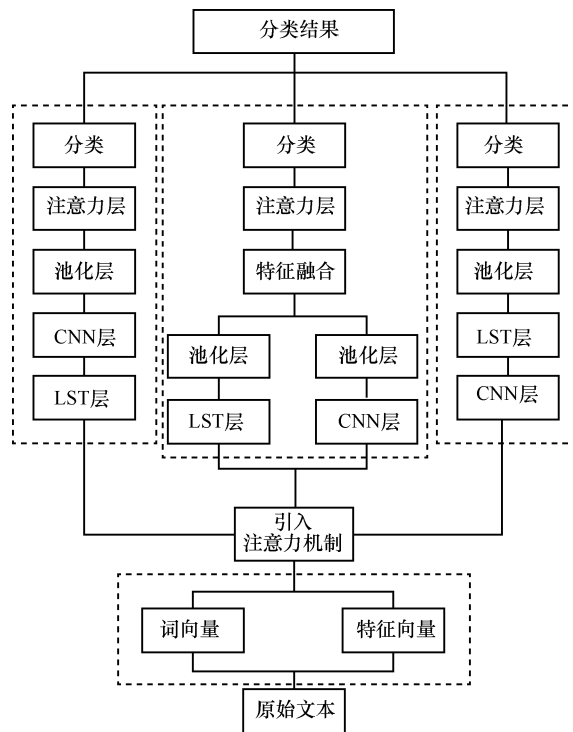


图 3 CNN 和 RNN 组合模型变种

该层的输入是一个矩阵  $M_r$ ，令  $M_r = \{r_1, r_2, \dots, r_{T_r}\}^T$ ， $r_i \in \mathbb{R}^{d_r}$ 。对于当前时刻  $t$ ，首先计算 3 个门单元，其依赖上一时刻的隐层状态向量  $h_{t-1}$  和当前输入  $r_t$ ，初始隐层状态向量为  $h_0$  为 0。

$$i_t = \sigma(W_{xi}r_t + W_{hi}h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{xf}r_t + W_{hf}h_{t-1} + b_f) \quad (2)$$

$$o_t = \sigma(W_{xo}r_t + W_{ho}h_{t-1} + b_o) \quad (3)$$

当前时刻的特征向量如下所示，同样依赖上一时刻的隐层状态向量  $h_{t-1}$  和当前输入  $r_t$ 。

$$g_t = \tanh(W_{xg}r_t + W_{hg}h_{t-1} + b_g) \quad (4)$$

紧接着，记忆单元整合前一时刻的记忆单元信息  $c_{t-1}$  和当前时刻的特征  $g_t$ ，并通过输入门保留当前时刻的某些信息以及遗忘门擦除前一时刻的某些信息。初始记忆单元  $c_0$  记为 0。

$$c_t = i_t \otimes g_t + f_t \otimes c_{t-1} \quad (5)$$

最终通过输出门得到隐层状态向量  $h_t$ 。

$$h_t = o_t \otimes \tanh(c_t) \quad (6)$$

上述公式中， $\sigma$  代表 sigmoid 激活函数， $\otimes$  代表按元素相乘。 $W$  和  $b$  代表相应的权重和偏置。

多个 LSTM 单元通过上面的方法就可以组合成一个 LSTM 网络，可以将  $\{c_1, c_2, \dots, c_{T_r}\}^T$  作为该网络输出的结果。

$$h_a = \{c_1, c_2, \dots, c_{T_r}\}^T \quad (7)$$

然而，对于一个输入序列，在一个时间节点  $t$ ，LSTM 网络只包含  $t$  以前的信息，却不包含  $t$  以后的信息。为了解决这个问题，本文采用双向 LSTM 的方法。该网络结构包含正向 LSTM 和反向 LSTM，正向 LSTM 按正常输入得到一个序列  $h_a$ 。反向 LSTM 将输入倒序，然后通过具有跟正向 LSTM 相同结构但权重参数不同的网络，最终也得到一个序列，再将这个序列倒序后得到  $h_b$ 。最后将 2 个序列相加得到  $H$ ，即为通过双向 LSTM 网络的最后结果。

$$H = h_a \oplus h_b \in \mathbb{R}^{T_r \times d_r} \quad (8)$$

### 3.3 卷积层

卷积层通过指定一定大小的窗口值提取某一类特征。该窗口值称为一个卷积核，卷积层可以有多个卷积核，卷积层的输入是一个矩阵  $M_c$ ，令  $M_c = \{m_1, m_2, \dots, m_{T_c}\}^T$  ( $m_i \in \mathbb{R}^{d_c}$ )。

假设卷积核滑动窗口大小为  $k \times d_c$ ，让该窗口在矩阵  $M_c$  上滑动，由于本文使用的是宽卷积操作，需先将矩阵  $M_c$  上下两侧各补零  $(k-1) \times d_c$  个，有

$$M_c = \{m_{2-k}, m_{3-k}, \dots, m_{T_c+k-1}\}^T, \quad m_i \in \mathbb{R}^{d_c}.$$

因此，当窗口在  $M_c$  上滑动时，将有  $k$  个长度为  $d_h$  的向量  $m_j, m_{j+1}, \dots, m_{j+k-1}$  落入第  $j$  个窗口 ( $2-k \leq j \leq T_c-1$ )。表示为

$$X_j = \{m_j, m_{j+1}, \dots, m_{j+k-1}\}^T \quad (9)$$

共有  $T_c + k - 1$  个窗口。假设卷积核为  $W$  ( $W \in \mathbb{R}^{k \times d_c}$ )，在每个窗口上进行点乘操作

$$Y_j = X_j W + b \quad (10)$$

最后卷积操作的结果为

$$C = \{Y_1, Y_2, \dots, Y_{T_c+k-1}\}^T \in \mathbb{R}^{(T_c+k-1) \times d_c} \quad (11)$$

### 3.4 池化层

池化层作用是将模型提取的特征进行过滤, 不仅能去除一些冗余信息, 还能减少网络的节点数进而减少训练参数数量。池化层的输入是  $M_p = \{p_1, p_2, \dots, p_{d_p}\} \in T_p \times d_p$ , 采用最大池化, 有

$$y_i = \max(p_i), i=1, 2, \dots, d_p \quad (12)$$

另外, 本文也采用了 Zeng 等<sup>[6]</sup>提出的一种分段池化的思想(如图4所示), 由于输入的句子被2个实体分割成3部分, 因此, 根据具体实体的位置, 将  $p_i$  分成3部分  $\{p_{i1}, p_{i2}, p_{i3}\}$ , 再分别对它们进行最大池化。

$$y_i = \{\max(p_{i1}), \max(p_{i2}), \max(p_{i3})\} \quad (13)$$

最后, 将  $y_i$  前后拼接起来得到

$$P = \{y_1, y_2, \dots, y_{d_p}\} \quad (14)$$

对于最大池化和分段最大池化的结果,  $P$  的大小分别为  $d_p \times 1$  和  $3d_p \times 1$ 。这种方法间接地将实体位置考虑进了模型, 实验结果也验证了分段池化的有效性。

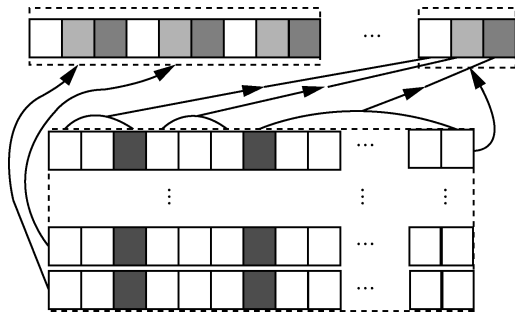


图4 分段池化

### 3.5 注意力机制

注意力层的目的是让模型能够自动识别输入句子中哪些成分对分类结果影响较大, 并对该成分赋予较大的权重, 以让模型增加对该成分的关注。注意力层可以加到大部分网络层之后, 甚至可以加到输入层之后重组词向量, 获得表现更为优异的组合词向量。

注意力层的输入为  $M_a$ ,  $M_a \in T_a \times d_a$ , 可将

$M_a$  展开成  $\{b_1, b_2, \dots, b_{T_a}\}^T$ , 通过一个待训练的权重参数  $\omega$  生成对每一个  $h_i$  的注意力权重。

$$M = \tanh(M_a) \quad (15)$$

$$\alpha = \text{softmax}(\omega^T M) \quad (16)$$

最后将每个  $\alpha_i$  乘以对应的  $b_i$ 。

$$R = M_a \alpha \quad (17)$$

通过注意力机制后得到的矩阵  $R$  拥有与输入相同的大小, 不同的是, 注意力机制通过注意力权重的加权对某些向量添加了“特别的关注”, 从而提高了模型的性能。

### 3.6 Dropout 和 L2 正则化

深度学习由于其参数数量众多, 普遍存在的一个问题是训练样本时容易过拟合。本文使用 dropout 和 L2 正则化来减少训练过程中的过拟合问题。dropout 首先由 Srivastava<sup>[12]</sup>提出。它在训练期间按照一定的概率屏蔽掉神经网络单元, dropout 层定义如下。

$$h_n = \omega_n^T (r \otimes x) + b_n \quad (18)$$

其中,  $r$  跟  $x$  具有相同的大小,  $r$  中的每一个元素以一定的概率  $p$  为 1, 以  $1-p$  的概率为 0, 这样 dropout 层就能以概率  $p$  保留神经网络单元,  $p$  越大, 保留的神经网络单元越多。

同样, L2 正则化也能减小过拟合问题, 本文在损失函数上添加了 L2 范数  $\|\theta\|_F^2$  项, 于是, 损失函数可以表示成

$$J(\theta) = -\sum_{i=1}^m t_i \log(y_i) + \lambda \|\theta\|_F^2 \quad (19)$$

第一项为交叉熵,  $\lambda$  为正则化系数。本文使用 Adam 优化算法<sup>[13]</sup>来更新参数。

### 3.7 卷积层和双向 LSTM 层的组合

无论是词向量、卷积层的输入输出还是 LSTM 层输入输出, 都是大小形如  $T \times d$  的矩阵, 因此可以随意组合两者, 本文采用了 3 种组合方式, 分别为 CNN 后接 LSTM (PCNN\_BLSTM)、LSTM 后接 CNN (BLSTM\_PCNN) 以及 CNN 与 LSTM (PCNN+BLSTM) 并联, 如图 1 所示。

## 4 实验

### 4.1 实验环境

实验环境与配置如表 1 所示。

表 1 实验环境与配置

实验环境	配置
操作系统	Win 10 64 位
CPU	Intel i7-6700HQ@2.6GHz (8CPU)
GPU	NVIDIA GeForce GTX 1060
内存	16 GB
编程语言	Python 3.6.3
深度学习框架	TensorFlow 1.5.0

## 4.2 数据集

本文使用的数据集是 SemEval-2010 task 8<sup>[1]</sup>, 该数据集已被广泛用于关系分类, 数据集包含 10 种关系, 包括代表实体之间没有关系的“Other”类别。数据集共包含 10 717 条语料, 其中, 8 000 条用来训练, 2 717 条用来测试, 每条句子中的实体以及实体之间的关系都已标记, 各种关系的分布如表 2 所示。

表 2 关系类别及其分布

关系类别	个数 (比例)
Other	1 410 (17.63%)
Cause-Effect	1 003 (12.54%)
Component-Whole	941 (11.76%)
Entity-Destination	845 (10.56%)
Product-Producer	717 (8.96%)
Entity-Origin	716 (8.95%)
Member-Collection	690 (8.63%)
Message-Topic	634 (7.92%)
Content-Container	540 (6.75%)
Instrument-Agency	504 (6.30%)

## 4.3 评价指标

针对每个关系类别, 一般用准确率 (Precision)、召回率 (Recall) 和  $F1$  值来衡量抽取结果, 对一个关系类别的判断, 分为 4 种。

TP (true positive): 将正确的判断正确。

FP (false positive): 将正确的判断错误。

FN (false negative): 将错误的判断错误。

TN (true negative): 将错误的判断正确。

基于这 4 种分类情况, 准确率和召回率定义如下。

$$Precision = \frac{TP}{TP + FP} \quad (20)$$

$$Recall = \frac{TP}{TP + FN} \quad (21)$$

$F1$  值则是准确率和召回率的一个调和平均值。

$$F1 = \frac{2Precision \cdot Recall}{Precision + Recall} \quad (22)$$

上面的指标仅仅是针对单个关系类别评估的, 为了评估模型在整个数据集上的性能, 采用每个类性能指标的宏平均 (macro-average)。

在一个关系分类中, 准确率表征的是所有模型判断为正确的句子中实际正确的所占的比例, 召回率表征的是测试集中所有该关系类别的句子中, 模型判断为正确所占的比例。简而言之, 准确率评估的是查准率, 召回率评估的是查全率。举个例子, 为什么传统基于规则的方法准确率较高而召回率低? 因为一般制定的规则都较为精确, 一个具有该规则的句子基本被判断正确, 因此准确率高。但是由于规则的不完备性, 很多本该属于这一类的句子没有被发现, 所以召回率低。

## 4.4 超参数设置

本文采用网络搜索的方法调整超参数, 将原始训练集中的  $\frac{1}{8}$  作为验证集, 剩余部分作为训练集, 在一定的可选超参数范围内 (部分经验参数参考文献[14-15]的设置), 进行多次实验, 进而优化模型参数, 同时为了便于对比, 不同模型中的相同结构采用相同的超参数。本实验的超参数设置如表 3 所示。

表 3 超参数设置

参数	值
词向量维度	300
位置向量维度	20
词性标注向量维度	60
批处理大小	128
adam 学习率	$1.00 \times 10^{-3}$
LSTM 层 dropout	0.3
其他 dropout	0.5
L2 正则化系数	$1.00 \times 10^{-4}$
LSTM 单元大小	512
CNN 卷积核窗口长度	3
CNN 卷积核个数	512

## 4.5 实验结果分析

本文实验将 CNN 和 LSTM 结合起来, 分别

采用了级联和并联共3种结合方式,如图3所示。为了确保实验的准确性,本文在相同输入下(词序列、位置信息、词性标注)复现了PCNN和BLSTM这2种模型,其中,PCNN模型方法参考文献[6],BLSTM模型方法参考文献[8]。同时实现了本文提出的3种组合模型变种,控制了一定的参数变量,使实验更具有可比性。表4展示了各模型的实验结果,结果表明,这3种方法均对原单一模型性能有了一定的提升,而且CNN比双向LSTM性能更优。

表4 各模型实验结果

模型	准确率	召回率	F1值
SVM <sup>[16]</sup>	82.25%	82.28%	82.19
PCNN	78.08%	87.26%	82.34
BLSTM	79.28%	82.61%	80.75
PCNN_BLSTM	82.32%	87.14%	<b>84.46</b>
BLSTM_PCNN	78.93%	84.73%	81.32
PCNN+BLSTM	81.25%	86.64%	83.78

另外,传统特征提取的方法,需要手动提取大量特征,以该类方法中性能最好的SVM模型为例,该模型需要提取包括词性标注、WordNet、依赖解析等12个人工特征,工作量巨大且耗时耗力。而本文提出的模型可以自动提取特征,并且表现更为优异,这也是神经网络的优势。同时,与现有的神经网络方法相比,本文将CNN和LSTM相结合,融合了单一模型的优点,既可以解决长句子中单词依赖问题,也可以提取更为丰富的特征,对单一模型均有提升。

为了验证模型的稳定性,每迭代一定次数,在测试集上验证数据,并统计F1值随迭代次数的变化,如图5所示。可见,各模型随着迭代次数的增加,F1值也逐渐增加,最后收敛趋于稳定,虽然有一定的波动,但波动不大,证明了模型的稳定性。

另外,本实验还统计了9个关系分类下各模型F1值的性能,如图6所示,可以发现,在特定关系类别下F1值差异明显,Cause-Effect和Entity-Destination明显优于其他类别,Instrument-Agency等表现不好,虽然在一定程度上,实验结果跟数据集的类别分布有一定的关系,但经过对测试数据集中2717条数据的分析与统计,还是可以发现数据中对于分类结果影响的差异。

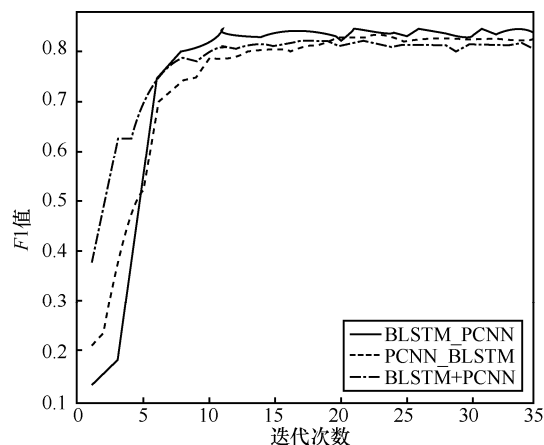


图5 F1值随迭代次数变化曲线

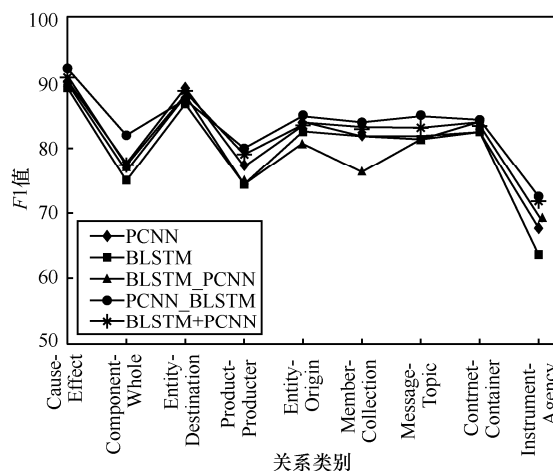


图6 各关系类别F1值比较

以模型PCNN\_BLSTM测试的结果为例,在F1值最高的Cause-Effect类别中,准确率达到92.99%,召回率达到90.27%,均达到了较高的水平。发现这些句子中往往会有高频词汇(如“cause”“result”)及其变体等出现,介词“by”“in”等也常常跟着“cause”“result”等一起出现,具有较为良好的结构特点,因此,这种类别性能表现出色。对于F1值最低的Instrument-Agency类别,本文统计了测试集中与该类别相关的句子,测试集中有156个句子表征该关系,模型预测正确了122个句子,而在所有的类别中,模型将179个句子预测为Instrument-Agency类,但其中只有122个句子是正确的。可见该类别的准确率和召回率分别为78.2%和68.15%,召回率相对较低。在预测正确的句子中,虽然也有高频词汇“use”及其变体出现,但还是有一大部分句子仅仅使用“by”“with”等介词来表示工具的使用,也就是说,介词有时候不伴随着高频词汇出现,而这些

介词往往为大部分句子所用，该类别句子结构特征较不明显，也是导致  $F1$  值相对较低的一个原因。当然，神经网络对关系的判别更为复杂，上面的分析也只是各类别  $F1$  值差异明显的其中一个可解释原因，神经网络对特征的提取是多方面的，甚至有不可解释的，人工难以提取的特征，这也是神经网络方法的优势之一。

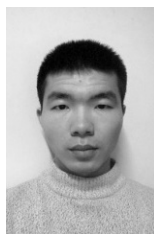
## 5 结束语

本文提出了一种融合 CNN 和双向 LSTM 的实体关系抽取方法。模型使用 3 种结合方式，考虑了不同模型结构对分类结果的影响，实验结果表明，在 SemEval-2010 task 8 数据集上取得了不错的效果。另外，由于句子的依存句法分析可以消除句中冗余信息的影响，下一步的工作会尝试加入依存句法结构等其他特征到模型的输入之中。

## 参考文献：

- [1] IRIS H. Semeval-2010 task 8: multi-way classification of semantic relations between pairs of nominals[C]//The Workshop on Semantic Evaluations: Recent Achievements and Future Directions. 2009.
- [2] 车万翔, 刘挺, 李生. 实体关系自动抽取[J]. 中文信息学报, 2005, 19(2): 2-7.  
CHE W X, LIU T, LI S. Automatic entity relation extraction[J]. Journal of Chinese Information Processing, 2005, 19(2): 2-7.
- [3] 黄瑞红, 孙乐, 冯元勇, 黄云平. 基于核方法的中文实体关系抽取研究[J]. 中文信息学报, 2008, 22(5): 102-108.  
HUANG R H, SUN L, FENG Y Y, et al. A study on kernel based chinese relation extraction[J]. Journal of Chinese Information Processing, 2008, 22(5): 102-108.
- [4] LIU C Y, SUN W B, CHAO W H, et al. Convolution neural network for relation extraction[C]//International Conference on Advanced Data Mining and Applications. 2013.
- [5] ZENG D J, et al. Relation classification via convolutional deep neural network[C]//COLING. 2014.
- [6] ZENG D J, LIU K, CHEN Y B, et al. Distant supervision for relation extraction via piecewise convolutional neural networks[C]//The 2015 Conference on Empirical Methods in Natural Language Processing. 2015.
- [7] ZHANG D X, WANG D. Relation classification via recurrent neural network[J]. arXiv preprint arXiv:1508.01006, 2015.
- [8] ZHANG S, ZHENG D Q, HU X C, et al. bidirectional long short-term memory networks for relation classification[C]//PACLIC. 2015.
- [9] ZHOU P, et al. Attention-based bidirectional long short-term memory networks for relation classification[C]//The 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2016.
- [10] TOMAS M, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space[J]. arXiv: 1301.3781, 2013.
- [11] SEPP H, SCHMIDHUBER J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.
- [12] NITISH S. Dropout: a simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958.
- [13] DIEDERIK K P, BA J. Adam: a method for stochastic optimization[J]. arXiv:1412.6980, 2014.
- [14] YAN X, et al. classifying relations via long short term memory networks along shortest dependency paths[C]//EMNLP. 2015.
- [15] WANG L L, ZHU C, et al. Relation classification via multi-level attention cnns[C]//The 54th Annual Meeting of the Association for Computational Linguistics. 2016.
- [16] BRYAN R, HARABAGIU S. Utd: classifying semantic relations by combining lexical and semantic resources[C]//The 5th International Workshop on Semantic Evaluation. 2010.

## [作者简介]



张晓斌 (1993-), 男, 福建漳州人, 国家数字交换系统工程技术研究中心硕士生, 主要研究方向为文本挖掘、实体关系抽取。



陈福才 (1974-), 男, 江西高安人, 硕士, 国家数字交换系统工程技术研究中心研究员、硕士生导师, 主要研究方向为大数据分析处理。



黄瑞阳 (1986-), 男, 福建漳州人, 国家数字交换系统工程技术研究中心助理研究员, 主要研究方向为文本挖掘、信息抽取。