计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(6):262−270 [doi: 10.15888/j.cnki.csa.007935]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于 BSTTC 模型的中文命名实体识别①

申　晖, 张英俊, 谢斌红, 赵红燕

(太原科技大学 计算机科学与技术学院, 太原 030024)
通讯作者: 申　晖, E-mail: 604554830@qq.com

摘　要: 大多数中文命名实体识别模型中, 语言预处理只关注单个词和字符的向量表示, 忽略了它们之间的语义关系, 无法解决一词多义问题; Transformer 特征抽取模型的并行计算和长距离建模优势提升了许多自然语言理解任务的效果, 但全连接结构使得计算复杂度为输入长度的平方, 导致其在中文命名实体识别的效果不佳. 针对这些问题, 提出一种基于 BSTTC (BERT-Star-Transformer-TextCNN-CRF) 模型的中文命名实体识别方法. 首先利用在大规模语料上预训练好的 BERT 模型根据其输入上下文动态生成字向量序列; 然后使用星型 Transformer 与 TextCNN 联合模型进一步提取句子特征; 最后将特征向量序列输入 CRF 模型得到最终预测结果. 在 MSRA 中文语料上的实验结果表明, 该模型的精确率、召回率和 $F1$ 值与之前模型相比, 均有所提高. 与 BERT-Transformer-CRF 模型相比, 训练时间大约节省了 65%.

关键词: BERT; 星型 Transformer; 命名实体识别; TextCNN; 条件随机场

引用格式: 申晖,张英俊,谢斌红,赵红燕.基于 BSTTC 模型的中文命名实体识别.计算机系统应用,2021,30(6):262–270. http://www.c-s-a.org.cn/1003-3254/7935.html

Chinese Named Entity Recognition Based on BSTTC Model

SHEN Hui, ZHANG Ying-Jun, XIE Bin-Hong, ZHAO Hong-Yan

(School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China)

Abstract: In most recognition models of Chinese named entities, language preprocessing only focuses on the vector representation of single words and characters and ignores the semantic relationship between them, hence failing to tackle polysemy. The transformer feature extraction model improves the understanding of natural language due to parallel computing and long-distance modeling, but its fully connected structure makes the computational complexity the square of the input length, which leads to poor recognition of Chinese named entities. A recognition method for Chinese named entities based on the BERT-Star-Transformer-TextCNN-CRF (BSTTC) model is proposed to solve these problems. First, the BERT model pre-trained on a large-scale corpus is used to dynamically generate the word vector sequence according to its input context. Then, the star Transformer-TextCNN model is adopted to further extract sentence features. Finally, the prediction result is received by inputting the feature vector sequence into the CRF model. The experimental results on the Chinese corpus from MSRA show that the accuracy, recall, and $F1$ value of this model are all higher than those of existing models. Moreover, its training time is 65% shorter than that of the BSTTC model.

Key words: BERT; Star-Transformer; named entity recognition; TextCNN; Conditional Random Fields (CRF)

① 基金项目: 山西省重点研发计划重点项目 (201703D111027); 山西省重点计划研发项目 (201803D121048, 201803D121055)
Foundation item: Key Project of Research and Development Program of Shanxi Province (201703D111027); Research and Development Project of Key Program of Shanxi Province (201803D121048, 201803D121055)
收稿时间: 2020-10-09; 修改时间: 2020-11-05; 采用时间: 2020-11-09; csa 在线出版时间: 2021-06-01

命名实体识别 (Named Entity Recognition, NER), 又称作“专名识别”, 是自然语言处理中的一项基础任务[1−3], 应用范围非常广泛. 命名实体一般指的是文本中具有特定意义或者指代性强的实体, 通常包括人名、地名、机构名、日期时间和专有名词等.

早期, 基于词典和规则的方法是命名实体识别任务中的主流方法, 但这种方法只能够在特定的语料上获得较高的识别效果, 而且费时费力、可移植性差, 在面对众多领域的复杂文本时, 该方法不再适用. 随着机器学习在自然语言处理领域的兴起[4−6], 将该方法应用于 NER 任务中成为一种新趋势. 在这种趋势下, 如何更好的解决序列标注问题成为提升命名实体识别效果的关键. 然而这种方法对特征选取的要求较高, 不仅需要从文本中选择对该项任务有影响的各种特征加入到特征向量中, 而且需要依据特定命名实体识别所面临的主要困难和所表现出的特性, 选择能有效反映该类实体特性的特征集合, 导致其通用性不佳, 泛化能力差. 近年来,由于分布式表示学习技术的蓬勃发展, 各种词向量表示方法层出不穷, 基于深度神经网络方法在 NER 这种典型的序列化标注问题上取得了较大进展.

1 相关工作

随着深度学习的快速发展, 源于神经网络模型的深度学习技术在 NER 任务中的表现越来越突出, 这种不依赖人工特征的端到端方案逐渐占据主流. 该方法对于 NER 问题的解决大致分为 3 个阶段: 通过学习嵌入模型, 以向量形式表示文本信息; 将以向量表示的文本输入到神经网络编码, 对文本序列建模; 最后解码层进行解码得到全局最优标注序列. 目前, 常用的生成词向量工具有 Mikolov 等提出的 Word2Vec 模型[7] 和 Pennington 等提出的 Glove 模型[8]. 但它们都无法解决多义词问题, 这两种模型对于不同语境下的词语产生的词向量是相同的, 这会对后续任务的结果产生影响. 谷歌于 2018 年提出了 BERT (Bidirectional Encoder Represe-ntations from Transformers) 模型[9], 该模型能够更深层次地提取文本的语义信息, 并且可以针对不同的上下文信息动态生成词向量, 并使 NLP 领域多个任务实验效果得到了大幅提升.

在序列标注任务当中, 常用的编码方式有循环神经网络 (Recurrent Neural Networks, RNN)、长短期记忆神经网络[10,11](Long Short-Term Memory, LSTM) 和卷积神经网络[12,13](Convolutional Nerual Networks, CNN). CNN 通过使用与字符向量维度相同的卷积核与字符向量组成的矩阵进行卷积得到其局部特征, 最后通过池化操作使得输出维度与输入维度保持一致. CNN 的优点在于可以利用 GPU 并行性快速提取局部特征, 缺点是很难使提取的字符特征包含全局信息. RNN 由于其具有良好的序列建模能力而常常被应用于命名实体识别任务中. 然而其缺点在于随着序列长度的增加, RNN 会逐步丧失学习能力, 出现“梯度消失”现象. 针对该问题, 有学者提出 RNN 的变体网络—LSTM. 通过添加门控机制缓解了“梯度消失”问题. 但由于它的循环结构无法利用 GPU 并行性, 这限制了它的计算效率. 为了解决 CNN 存在的无法捕获全局信息与 RNN 运算效率低下的问题, 谷歌于 2017 年提出了具有更强大特征抽取能力的 Transformer 编码器模型, 并在多个 NLP 任务中取得了良好的结果. 但由于 Transformer 模型[14] 的结构为全连接结构, 所以它的计算和内存开销是句子长度的平方倍, 参数量也较大, 需要较长的训练时间. 而在解码阶段, 常用的模型有 *Softmax*、条件随机场 (Conditional Random Field, CRF). 其中, 条件随机场模型是目前解决序列标注问题的最为经典的方法. 因为该模型充分考虑了标签与前后文标注的关系, 所以能够较好地解决标注偏置等问题.

由于 LSTM 在处理时间序列数据时可以很好地获取和保存序列的上下文信息, 目前 LSTM-CRF 已成为 NER 任务的基础网络架构之一, 许多研究人员尝试在其基础上添加各种相关特征来提高最终的识别效果. 例如 Lample 等[15] 于 2016 年提出 BiLSTM-CRF 模型, 该模型使用双向 LSTM 提取字符特征, 并取得了当时最好的识别效果; Huang 等[16] 在 BiLSTM-CRF 基础上加入手工拼写特征; Ma 等[17] 在预训练好的词向量中融入了字符级 CNN 抽取的特征; 而 Chiu 等[18] 还加入了多种预训练好的词典特征. 上述这些方法中使用的初始向量表示都是通过随机生成或 Word2Vec 预训练语言模型产生, 导致其识别效果并未达到最好. 也有基于 CNN 的命名实体识别方案, 例如 Collobert 等[19] 提出了 CNN-CRF 网络结构; Santos 等[20] 又扩展了该网络结构, 在其基础上添加卷积层提取字符级特征; Strubell 等[21] 首次提出了空洞卷积网络 (IDCNN) 来提取特征, 扩大了感受野的同时减少了参数数量. 由于以上方法使用 CNN 为基本结构提取特征无法充分获取全局信

息, 所以其识别效果还有待提高.

以上所述方法都存在共同的问题: 初始嵌入无法表示一词多义. 由于 BERT 可以充分表征不同语境中的句法与语义信息, 近几年, 开始有研究人员考虑使用 BERT 模型来生成初始嵌入, 例如: Straková等[22] 将 BERT 模型应用在嵌套命名实体识别中, 提升了识别效果; 谢腾等[23] 采用了 BERT-BiLSTM-CRF 模型进行中文命名实体识别, 在 MSRA 数据集上达到了较高 $F1$ 值 94.65%; 李妮等[24] 提出基于 BERT-IDCNN-CRF 的中文命名实体识别方法,该方法通过 BERT 预训练语言模型得到字的上下文表示, 再将字向量序列输入 IDCNN-CRF 模型中进行训练. 虽然这些方法使用了 BERT 模型得到文本向量表示, 但在特征抽取速度和效果上还需进一步提高.

近年来, 随着中文命名实体识别的效果不断提高, 将命名实体方法应用于某个特定领域成为了一个新的研究热点. 例如: 李丽双等[25] 为了抽取出生物医学语料中的相关命名实体, 提出了 CNN-BiLSTM-CRF 网络模型, 并得到了较好的效果; 周晓磊等[26] 针对财产纠纷审判案件文书提出 SVM-BiLSTM-CRF 模型, 首先利用 SVM 筛选出关键句子, 并将其以字符向量表示, 输入 BiLSTM-CRF 模型中抽取出动产、不动产、知识财产 3 类实体; 杨文明等[27] 提出了 IndRNN-CRF 和 IDCNN-BiLSTM-CRF 模型, 并将其应用于医疗文本中的命名实体抽取任务中, 使得该模型在 $F1$ 值和精确率上都优于经典的 BiLSTM-CRF 模型.

为了解决一词多义问题, 并且可以在提高特征抽取速度的同时保证模型的识别效果, 本文提出了一种基于 BSTTC 模型的中文命名实体识别方法, 使用 BERT 动态生成句子的表示矩阵, 将该矩阵输入联合模型中进一步抽取特征, 最后由 CRF 模型得到最佳预测序列. 实验结果表明, 模型在 MSRA 数据集上的 $F1$ 值达到了 95.69%. 与 BERT-Transformer-CRF 模型相比, 训练时间大约节省了 65% 的时间.

2 BSTTC 模型

模型主要由 3 个模块构成, 分别是语言表示模块、特征抽取与融合模块以及标签解码模块, 其整体结构如图 1 所示. 模型首先利用 BERT 预训练语言模型将标注语料动态表示为含有上下文语义信息的字符向量序列; 然后将其分别输入具有轻量结构的星型 Transformer 模型与 TextCNN 模型中进一步提取局部特征与全局特征; 接着将两种特征进行融合得到新的向量序列; 最后将经过特征融合后的向量输入 CRF 层进行解码, 得到每个字符的标签类别.

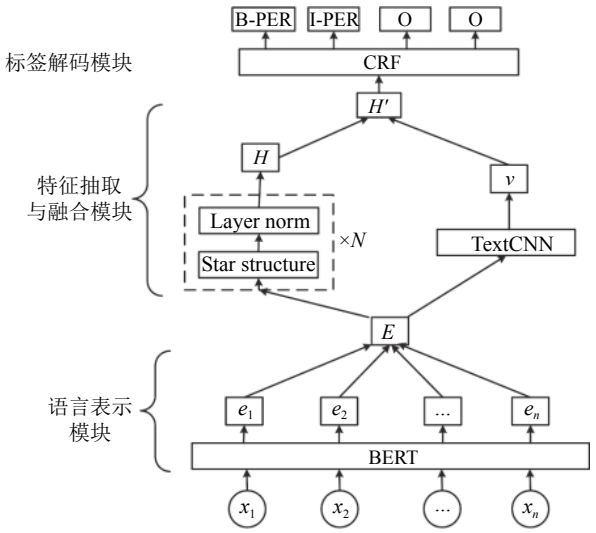

图 1 BSTTC 模型结构

与现有的中文命名实体识别方法相比, 本文提出的方法优势在于: ① 利用 BERT 预训练语言模型动态得到了含有丰富语义信息的句子表示, 解决了一词多义的问题; ② 使用了星型 Transformer 模型和 TextCNN 分别提取局部特征和全局特征, 将其进行融合, 使得每一个字符向量既具有句子表示又具有字符级表示; ③ 星型 Transformer 模型在 Transformer 模型的基础上优化了网络结构, 大大减少了参数数量, 缩短了训练时间, 同时提高了 $F1$ 值.

2.1 BERT 预训练语言模型

词嵌入技术是为了将自然语言中的词映射到一个低维度稠密的连续向量空间中, 使得语义相似的词可以共享上下文信息, 从而提升泛化能力. 但是传统的词嵌入学到的是一个词的固定语义, 无法解决一词多义问题. 针对该问题, 本文采用了谷歌发布的中文 BERT 预训练语言模型.

BERT 预训练语言模型采用双向 Transformer 作为特征抽取器, 完全基于多头自注意力机制对一段文本进行建模, 可以无损失捕获更长的上下文信息, 提高了特征抽取能力. 同时, 使用“Masked 语言模型”无监督预测任务捕捉词级别表示, 充分利用词左右上下文信息获得更好的词分布式表示. 该任务使用随机遮挡方法

为 BERT 模型赋予了一定的文本纠错能力, 而且缓解了 finetune 时候与预训练时输入不匹配的问题 (预训练时输入句子当中有 mask, 而 finetune 时的输入是完整的句子, 即为输入不匹配问题).

在中文命名实体识别任务中, BERT 的输入为单个句子. 句子中每个字符对应 3 个向量, 其中, Token Embeddings 为字符向量, 用于下游的分类任务; Segment Embeddings 为分段向量, 在句子对任务中用于区分不同句子; Position Embeddings 为位置向量, 用于得到每个字符在序列中的相对位置信息.

通过使用 BERT 预训练语言模型, 最终得到一个由字符嵌入序列组成的句子矩阵$E \in R^{n\times d}$, 矩阵中的一行代表一个字符向量. 所以, 一个由 n 个字符组成的句子$X=\{x_1,x_2,\cdots,x_n\}$可以被表示为: $E=[e_1,e_2,\cdots,e_n]$, 其中 e_m是第m个字符嵌入.

2.2 星型 Transformer 模型

Transformer 模型由于其独特的结构组合, 在自然语言处理任务中表现出了良好的特征抽取能力. 但由于 Transformer 模型的结构为全连接结构, 如图 2 所示, 所以它的计算和内存开销是句子长度的平方倍, 参数量较大, 导致模型的训练需要较长时间. 针对该问题, 本文提出使用 Transformer 模型的变体—星型 Transformer 模型提取句子特征, 该模型具有轻量级的结构, 核心思想是通过将完全连接的拓扑结构变换成星形结构来稀疏架构. 模型结构[28] 如图 3 所示.

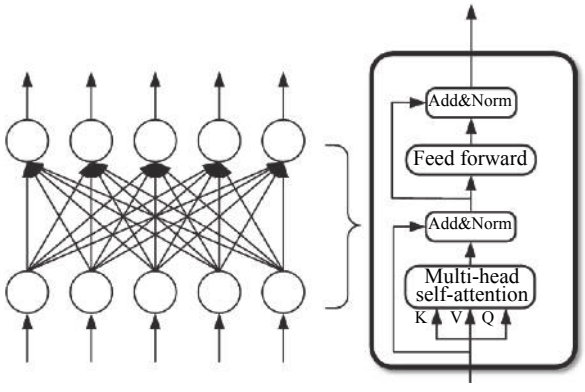

图 2 Transformer 模型结构

在图 3 星型 Transformer 模型中, 包含两种结点: 一个中心结点和 n 个卫星结点. 每个卫星结点之间以及卫星结点与中心结点之间都存在信息的传递. 其中, 卫星结点之间的连接使得每个卫星节点从其相邻结点收集信息; 卫星结点与中心结点的连接可以使得每两个非相邻的卫星节点可以通过中心结点进行信息传递.

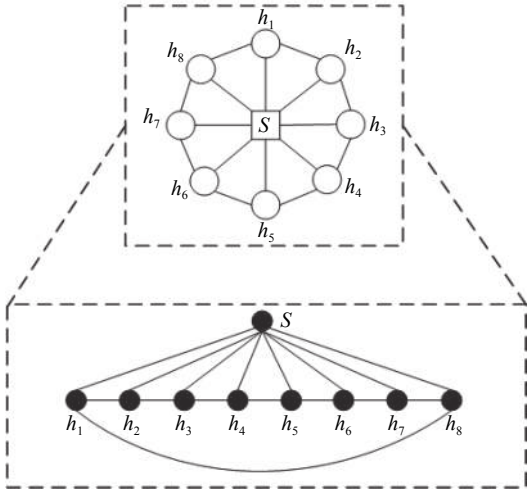

图 3 星型 Transformer 模型结构

与 Transformer 中的建模机制相同, 星型 Transformer 中每个结点的状态同样基于多头自注意力机制进行更新, 其中, 自注意力机制过程如式 (1) 所示.

$$Attention(Q,K,V)=Softmax\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V \tag{1}$$

在自注意力机制中, 每个初始字符嵌入向量乘以 3 个不同的权值矩阵w_q、w_k、w_v, 从而得到 3 个维度相同的向量, 分别为 Query 向量 (Q)、Key 向量 (K) 和 Value 向量 (V). QK^{T}计算出每个字向量之间的紧密程度得分, 然后除以一个惩罚因子$\sqrt{d_k}$, 使得 Q、K 的内积保持在一个合理范围内. 接着使用 *Softmax* 对其进行归一化处理得到 *Attention* 值, 并与 *Value* 向量相乘, 最后输出所有字符向量的带权和, 使得每个新的字符向量都包含了其余每个字符的信息.

由于事物具有多面性, 而自注意力机制只能关注到单方面的信息, 为使模型能够同时关注到来自不同位置与不同子空间的信息, 星型 Transformer 同样采用了“多头”模式, 既将每个头得到的信息进行拼接, 将拼接后得到的矩阵转换为一个新的向量, 如式 (2)、式 (3) 所示.

$$Multihead=concat(head_1,\cdots,head_n)\cdot W \tag{2}$$

$$head_i=Attention\left(QW_q^i,KW_k^i,VW_v^i\right) \tag{3}$$

2.2.1 卫星结点的更新

当使用星型 Transformer 编码长度为n的文本序列时, 设它的初始嵌入矩阵为: $E \in R^{n\times d}$, 所有卫星结点与

中心结点更新一次为一步更新. 假设在 t 步更新后, 中心节点的状态为$s^t \in R^{1\times d}$, 所有n个卫星节点的状态为(字符维度设为 d 维)$H^t = \left[h_1^t,\cdots,h_n^t\right]$, $H^t \in R^{n\times d}$.

初始化$H^0 = E$, $s^0 = average(E)$.

在第 t 步更新时, 每个卫星节点与其上下文做多头注意力, 其上下文信息包括序列中的相邻节点h_{i-1}^{t-1}、h_{i+1}^{t-1}、中心节点s^{t-1}、该结点先前状态h_i^{t-1}与其对应的字符嵌入, 更新过程如式 (4)、式 (5) 所示:

$$C_i^t = \left[e_i; s^{t-1}; h_{i-1}^{t-1}; h_i^{t-1}; h_{i+1}^{t-1}\right] \tag{4}$$

$$h_i^t = MultiAtt\left(C_i^t, h_i^{t-1}\right) \tag{5}$$

在信息交换之后, 对每个卫星结点进行层归一化操作, 如式 (6) 所示:

$$h_i^t = LayerNorm\left(ReLU\left(h_i^t\right)\right),\ i \in [1,n] \tag{6}$$

2.2.2 中心结点的更新

在第 t 步更新时, 所有卫星结点更新之后, 中心结点与所有更新后的卫星节点H^t及其先前状态s^{t-1}做多头注意力, 然后进行层归一化操作, 更新过程如式 (7)–式 (9) 所示:

$$C_i^t = \left[H^t; s^{t-1}\right] \tag{7}$$

$$s^t = MultiAtt\left(C_i^t, s^{t-1}\right) \tag{8}$$

$$s^t = LayerNorm\left(ReLU\left(s^t\right)\right) \tag{9}$$

最终, 通过多步更新卫星和中心结点, 星型 Transformer 模型最终得到新的句子矩阵: $H = [h_1, h_2, \cdots, h_n]$, $H \in R^{n\times d}$. 其整体更新过程如算法 1 所示.

算法 1. 星型 Transformer 整体更新算法

输入: $E=[e_1,e_2,\cdots,e_n]$

输出: $H=[h_1,h_2,\cdots,h_n]$

1. //初始化
2. $h_1^0,\cdots,h_n^0 \leftarrow e_1,\cdots,e_n$
3. $\mathbf{s}^0 \leftarrow average(\mathbf{e}_1,\cdots,\mathbf{e}_n)$
4. for t 1 to T do
5. //更新全部卫星结点
6. for i 1 to n do
7. $C_i^t=\left[e_i;s^{t-1};h_{i-1}^{t-1};h_i^{t-1};h_{i+1}^{t-1}\right]$
8. $h_i^t=MultiAtt(C_i^t,h_i^{t-1})$
9. $h_i^t=LayerNorm(ReLU(h_i^t)),i\in[1,n]$
10. //更新中心结点
11. $C_i^t=[H^t;s^{t-1}]$
12. $s^t=MultiAtt(C_i^t,s^{t-1})$
13. $s^t=LayerNorm(ReLU(s^t))$
14.//输出由卫星结点状态组成的句子矩阵: $H=[h_1,h_2,\cdots,h_n]$

2.3 TextCNN 模型

由于星型 Transformer 模型改变了 Transformer 模型中的全连接结构, 使得信息传递过程局限于邻近结点, 无法像全连接结构一样充分提取句子的全局信息. 鉴于卷积操作可以充分利用 GPU 并行性, 基于该问题, 本文提出使用 TextCNN 模型[29] 提取句子特征, 得到含有全局信息的句子向量.

该模型结构如图 4 所示, 图中文本矩阵由 BERT 预训练语言模型产生的字符嵌入向量组成, 卷积层的过滤器大小分别为 3、4、5、6. 在卷积层使用不同的卷积核由上往下滑动与矩阵做卷积操作, 卷积核的宽度和字符向量的维度一致, 每个卷积核获得一列 feature map. 卷积过程如式 (10)、式 (11) 所示:

$$c_i = f(w \cdot e_{i:i+h-1} + b) \tag{10}$$

$$c = [c_1, c_2, \cdots, c_{n-h+1}] \tag{11}$$

其中, $e_{i:i+h-1} \in R^{h\times d}$表示由字符嵌入序列$e_i, e_{i+1}, \cdots, e_{i+h-1}$组成的矩阵, $w \in R^{h\times d}$是卷积核, f 是非线性函数, b 是偏置, c 为卷积核 w 获得的 feature map.

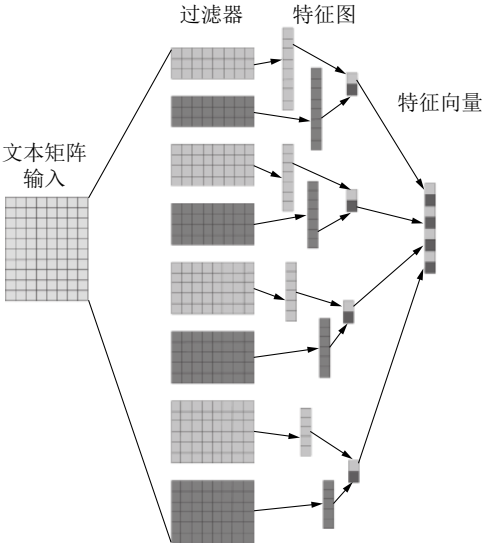

图 4 TextCNN 模型

每个 feature map 通过 max-pooling 都会得到一个特征值, 这个操作也使得 TextCNN 能处理不同长度的文本. 连接每个特征值形成一个一维向量作为含有 Dropout 层的全连接层的输入, 经过激活函数输出. 并在全连接层上添加 L2 正则化参数. 最后将全连接层的输出使用 *Softmax* 函数, 获取文本分到不同类别的概率. 本文中旨在使用 TextCNN 模型得到该句子的全局

特征, 所以丢掉最后一层. 最终该模型的输出为一维句子向量: $v \in R^{1\times d}$.

在中文命名实体识别任务中, 字符的标签判别过程不仅要考虑该字符周围的信息, 即局部特征, 句子中包含的全局信息也有助于最终的标签预测, 所以, 融合局部特征和全局特征是有必要的. 目前, 常用的特征融合策略有两种: concat 和 add. 由于 concat 是通过将向量拼接来融合信息, 维度增加, 最终导致计算量的增加, 所以, 本文采用了 add 融合策略得到最终的文本表示矩阵, 即: 将 TextCNN 与 Star-Transformer 模型的输出进行融合: $H' = [(h_1 + v), \cdots, (h_n + v)]$, $H' \in R^{n\times d}$.

2.4 CRF 模型

命名实体识别本质上是一种多分类问题, 所以在解码阶段 *Softmax* 分类器是一种常用的方法. 但由于该方法只是单纯的分类, 没有考虑到标签之间含有依存关系. 因此, 本文使用条件随机场模型 (CRF). CRF 是给定一组输入序列条件下另一组输出序列的条件概率分布模型, 在自然语言处理中得到了广泛应用.

在 CRF 中, 每个句子$X = \{x_1, x_2, \cdots, x_n\}$都有一个待选标签序列集合 Y_X, 通过计算集合中每个标签序列 $Y = \{y_1, y_2, \cdots, y_n\}$的得分来决定最终的标注序列, 计算得分过程如式 (12) 所示.

$$score(x,y) = \sum_{i=1}^{n} P_{i,y_i} + \sum_{i=1}^{n+1} A_{y_{i-1},y_i} \tag{12}$$

其中, $P \in R^{n\times k}$是一个得分矩阵, k 为所有标签数量, $P_{i,j}$表示句子中第 i 个字符对应第 j 个标签的分数; $A \in R^{(k+2)\times(k+2)}$是一个包含了句子开始与结束标签的转移矩阵, $A_{i,j}$则表示标签 i 到标签 j 的转移分数.

最后将每个标签序列的分数进行归一化得到概率, 其中概率最大的标签序列即为该句子的最终标注序列, 归一化过程如式 (13) 所示.

$$P(y \mid x) = \frac{\exp(score(x,y))}{\sum_{y' \in Y_X} \exp(score(x,y'))} \tag{13}$$

3 实验及结果分析

3.1 实验环境

本文所做实验均在 Ubuntu 操作系统上进行; 处理器为 i7-6700HQ@2.60 GHz; 内存大小 16 GB; 显存大小为 10 GB; 使用深度学习框架 PyTorch 1.2.0 构建所有神经网络模型进行训练和测试; 使用 Python 3.6 编程语言进行代码编写.

3.2 实验数据

本文采用微软亚洲研究院公开的 MSRA 数据集进行实验. 该数据集中含有训练集与测试集, 包含的实体类型有人名、机构名、地名. 其中, 训练集和测试集分别由 46 400 个句子和 4400 个句子组成. 数据集中各类实体统计如表 1 所示.

表 1 数据集实体个数

数据集	地名	机构名	人名	共计
训练集	36 517	20 571	17 615	74 703
测试集	2877	1331	1973	6181

3.3 标注策略与评价指标

在命名实体识别任务中, 有 BOI、BOIE、BOIES 三种标注方法. 本文采用了 BOI 标注策略, 其中实体中第一个字符用“B”代表, “O”表示该字符为非实体, 实体中第一个字符以外的字符用“I”表示. 所以, 将实体边界与实体类型结合可以得到 7 种待预测标签: “O”, “B-PER”, “B-LOC”, “B-ORG”, “I-PER”, “I-LOC”和“I-ORG”.

在命名实体识别任务中, 精确率 P、召回率 R 和 $F1$ 值是常用的 3 种评价指标. 每种评价指标的具体计算过程如公式 14 所示. 其中, T_P为预测出是实体并预测正确的个数, F_P为预测出为实体但预测错误的个数, F_N为是实体但预测为非实体的个数.

$$\begin{cases} P = \dfrac{T_P}{T_P + F_P} \times 100\% \\ R = \dfrac{T_P}{T_P + F_N} \times 100\% \\ F1 = \dfrac{2PR}{P + R} \times 100\% \end{cases} \tag{14}$$

3.4 参数设置

本实验使用 BERT-Base 预训练语言模型作为向量表示层, 该模型共有 12 层, 在多头注意力中头数为 12, 隐层输出为 768 维, 参数大小为 110 MB. 星型 Transformer 模型的层数分别设为 1、2、3、4 层, TextCNN 中采用单通道方式, 由于数据集中实体最大长度为 6, 所以卷积核设置四种不同的尺寸, 宽度与字符向量维度一致, 高度分别为 3、4、5、6. 具体网络训练参数设置如表 2 所示.

3.5 实验过程及结果分析

在实验中, 首先验证了星型 Transformer 模型层数对 $F1$ 值的影响. 随着训练迭代次数增加, BSTTC 模型的 $F1$ 值变化如图 5 所示, 其中, 每条折线代表了不同

星型结构层数时模型的 $F1$ 值变化. Star-Transformer-1 表示星型结构的层数为 1 层, 其他模型以此类推. 实验中其余超参数不变, 只改变星型结构层数. 实验表明, 效果最好的是 Star-Transformer-3 模型, 并在第 16 个 epoch 时 $F1$ 值达到最大 95.69%.

表 2 参数配置

参数	值
char emb dim	768
max seq_length	128
学习速率	1e–5
star_dropout	0.1
TextCNN_dropout	0.5
Star-Transformer layer	1, 2, 3, 4
filter height	3, 4, 5, 6

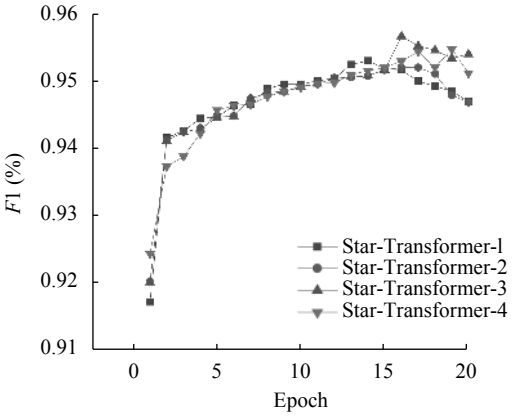

图 5 星型结构模型不同层数的 $F1$ 值

表 3 中分别列出了取得最大 $F1$ 值时数据集中每种实体识别的准确率、召回率和 $F1$ 值. 与人名和地名实体相比, 机构类实体的识别效果较差, 原因可能在于大部分机构名中都嵌套有地名, 这对于最终的预测造成了较大的干扰, 导致预测效果不佳.

表 3 BSTTC 不同类型命名实体识别结果 (%)

类型	P	R	$F1$
LOC	94.85	94.59	95.38
ORG	94.36	92.87	93.26
PER	96.72	96.47	96.73
ALL	95.89	94.86	95.69

为了验证星型 Transformer 模型轻量结构的优越性, 还在该语料上与 BERT-Transformer-CRF 模型进行了对比, 对比结果如图 6 所示. 可以看出, BSTTC 模型的收敛速度更快, 在训练初期, 就能够达到一个较高的 $F1$ 值, 并且持续提升, 最后保持在一个相当高的水平上. 而 BERT-Transformer-CRF 模型在多次迭代更新后才会上升到一个较高水平, 但还是无法超过 BSTTC 模型.

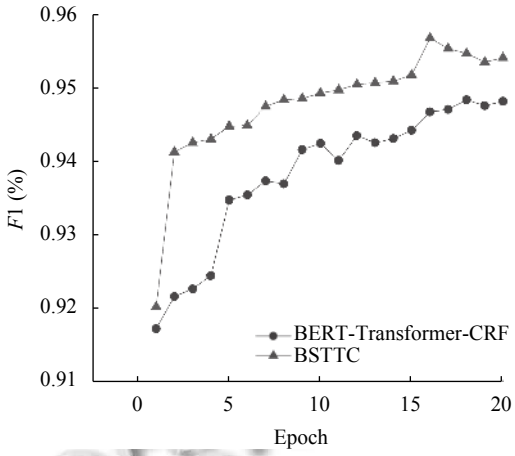

图 6 实验结果对比

表 4 中分别列出了 BERT-Transformer-CRF 和 BSTTC 模型迭代训练的累计时间及其对应的 $F1$ 值. 可以看到, BERT-Transformer-CRF 模型在第 18 个 epoch 时得到最优 $F1$ 值 94.85%, 而 BSTTC 模型在第 16 个 epoch 时就达到最大 $F1$ 值 95.69%, 此时它们的训练时间分别为 19238 s 与 54725 s, 与 BERT-Transformer-CRF 模型相比, BSTTC 的训练时间大约节省了 65%.

表 4 迭代训练累计时间

Epoch	BERT-Transformer-CRF		BSTTC	
	训练时间(s)	$F1$(%)	训练时间(s)	$F1$(%)
1	3968	91.75	1286	92.04
2	6923	92.18	2478	94.14
3	9916	92.29	3675	94.27
4	12876	92.47	4868	94.32
5	15786	93.49	6064	94.49
6	18762	93.56	7254	94.51
7	21731	93.75	8456	94.76
8	24728	93.71	9655	94.85
9	27739	94.18	10848	94.87
10	30736	94.26	12045	94.94
11	33724	94.03	13249	94.99
12	36719	94.36	14443	95.06
13	39706	94.27	15637	95.08
14	42727	94.32	16841	95.10
15	45726	94.44	18037	95.19
16	48728	94.69	**19238**	**95.69**
17	51739	94.72	20442	95.54
18	**54725**	**94.85**	21641	95.48
19	57683	94.77	22836	95.36
20	60692	94.83	24029	95.42

此外, 为了验证模型的有效性, 本文还在该语料上与以下模型进行了对比:

1) Radical-BiLSTM-CRF 模型, 由 Dong 等[30] 提出. 该模型将字的嵌入和笔画表示的连接输入到 BiLSTM-

CRF 中进行训练.

2) Lattice-LSTM-CRF 模型, 由 Zhang 等[31] 提出, 该模型在嵌入层利用注意力机制融合了字符与词粒度特征, 其中单词选取原则为该字符居于单词末位.

3) DEM-attention 模型, 由 Zhang 等[32] 提出, 该模型同样利用注意力机制在嵌入层中动态结合了字符和单词粒度的特征, 只是单词选取原则稍有不同, 该字符在句子中对应的所有单词都包含在内, 然后将其输入 BiLSTM-CRF 中进行训练.

4) BERT-BiLSTM-CRF 模型, 该模型采用预训练好的 BERT 模型产生字向量, 输入 BiLSTM-CRF 模型中进行训练.

5) CAN 模型, 由 Zhu 等[33] 提出, 该模型将预训练好的词向量输入 CNN 和 GRU 网络从相邻字符和句子上下文中捕获信息, 并使用了 CRF 进行标签预测.

6) BERT-Transformer-CRF 模型, 该模型类似于 BERT-BiLSTM-CRF 模型, 将 BiLSTM 层替换为 Transformer 层.

7) BERT-Star-Transformer-CRF 模型, 该模型类似于 BERT-BiLSTM-CRF 模型, 将 BiLSTM 层替换为 Star-Transformer 层.

表 5 中分别列出了每种模型的精确率、召回率和 $F1$ 值实验结果.

表 5 与其它模型对比结果 (%)

序号	模型	P	R	$F1$
1	Radical-BiLSTM-CRF[30]	91.39	88.22	89.78
2	Lattice-LSTM-CRF[31]	93.57	92.79	93.18
3	DEM-attention[32]	90.59	91.15	90.87
4	BERT-BiLSTM-CRF	92.84	94.57	93.68
5	CAN[33]	93.53	92.42	92.97
6	BERT-Transformer-CRF	94.57	95.15	94.85
7	BERT-Star-Transformer-CRF	93.48	96.37	95.54
8	BSTTC	**94.79**	**96.84**	**95.69**

从对比结果可以看出, 与其它模型相比, BSTTC 模型在精确率、召回率和 $F1$ 值 3 方面均有提高.

1) 将模型 4 与模型 1、模型 2、模型 3、模型 5 作对比, 可以发现模型 4 的 $F1$ 值最高, 说明 BERT 抽取的特征比单独训练笔画特征和字词融合特征更丰富, BERT 字向量更好的结合了上下文, 可以更好的表示字的语义信息.

2) 将模型 6 与模型 4 做对比, 可以发现与 BiLSTM 相比, Transformer 模型的特征抽取能力更强, 可以得到具有更丰富语义信息的字符特征.

3) 将模型 6、模型 7 对比, 可以发现在召回率和 $F1$ 值上都有一定程度的提高, 在精确率上有所下降, 说明星型 Transformer 模型在简化结构的同时保留了绝大部分捕获长期依赖的能力.

4) 将模型 7、模型 8 做对比, 加入 TextCNN 模型后, 精确率、召回率和 $F1$ 值都有所提高, 且都高于 BERT-Transformer-CRF 模型, 充分表明了与 Transformer 模型捕获的特征相比, TextCNN 捕获的全局特征与星型 Transformer 模型融合后的特征更加丰富, 更有助于标签的判别.

4 结束语

针对传统词向量表示方法无法表征字多义性, 以及 Transformer 特征抽取模型参数量大, 训练时间长, 无法充分提取全局信息的问题, 提出了基于特征融合的 BSTTC 模型. 该模型摒弃了传统语言模型的缺点, 使用 BERT 动态生成含有丰富语义特征与语法结构特征的字符向量, 然后通过星型 Transformer 与 TextCNN 联合模型进一步提取特征, 在减少训练时间的同时保证了特征抽取能力. 结果表明, 与以往模型相比, 本文的 BSTTC 模型在 MSRA 数据集上取得了最好的效果. 下一步将考虑引入外部信息, 提升复杂嵌套实体的识别效果.

参考文献

1 张晓艳, 王挺, 陈火旺. 命名实体识别研究. 计算机科学, 2005, 32(4): 44 –48. [doi: 10.3969/j.issn.1002-137X.2005.04.014]

2 张涛, 贾真, 李天瑞, 等. 基于知识库的开放领域问答系统. 智能系统学报, 2018, 13(4): 557–563.

3 庞亮, 兰艳艳, 徐君, 等. 深度文本匹配综述. 计算机学报, 2017, 40(4): 985–1003.

4 曲春燕. 中文电子病历命名实体识别研究 [硕士学位论文]. 哈尔滨: 哈尔滨工业大学, 2015.

5 王鹏远, 姬东鸿. 基于多标签 CRF 的疾病名称抽取. 计算机应用研究, 2017, 34(1): 118–122. [doi: 10.3969/j.issn.1001-3695.2017.01.025]

6 李业刚, 黄河燕, 鉴萍. 引入混合特征的最大名词短语双向标注融合算法. 自动化学报, 2015, 41(7): 1274–1282.

7 Ma L, Zhang YQ. Using Word2Vec to process big text data. Proceedings of 2015 IEEE International Conference on Big Data. Santa Clara, CA, USA. 2015. 2895–2897.

8 Sharma Y, Agrawal G, Jain P, *et al*. Vector representation of words for sentiment analysis using GloVe. Proceedings of

2017 International Conference on Intelligent Communication and Computational Techniques. Jaipur, India. 2017. 279–284.
9 Devlin J, Chang MW, Lee K, *et al*. Bert: Pre-training of deep bidirectional transformers for language understanding. Proceedings of 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis, MN, USA. 2019. 4171–4186.
10 Peng NY, Dredze M. Improving named entity recognition for chinese social media with word segmentation representation learning. arXiv: 1603.00786, 2016.
11 He HF, Sun X. A unified model for cross-domain and semi-supervised named entity recognition in chinese social media. Proceedings of the 31st AAAI Conference on Artificial Intelligence. San Francisco, CA, USA. 2017. 3216–3222.
12 Chen H, Lin ZJ, Ding GG, *et al*. GRN: Gated relation network to enhance convolutional neural network for named entity recognition. Proceedings of the 33rd AAAI Conference on Artificial Intelligence. Honolulu, HI, USA. 2019. 6236−6243.
13 Gehring J, Auli M, Grangier D, *et al*. Convolutional sequence to sequence learning. Proceedings of the 34th International Conference on Machine Learning. Sydney, Australia. 2017. 1243–1252.
14 Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach, NY, USA. 2017. 6000–6010.
15 Lample G, Ballesteros M, Subraman S, *et al*. Neural architectures for named entity recognition. Proceedings of 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego, CA, USA. 2016. 260–270.
16 Huang ZH, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. arXiv: 1508.01991, 2015.
17 Ma XZ, Hovy E. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin, Germany. 2016. 1064–1074.
18 Chiu JPC, Nichols E. Named entity recognition with bidirectional LSTM-CNNs. Transactions of the Association for Computational Linguistics, 2016, 4: 357–370. [doi: 10.1162/tacl_a_00104]
19 Collobert R, Weston J, Bottou L, *et al*. Natural language processing (almost) from scratch. The Journal of Machine Learning Research, 2011, 12: 2493–2537.
20 dos Santos C, Guimarães V. Boosting named entity recognition with neural character embeddings. Proceedings of the 5th Named Entity Workshop. Beijing, China. 2015. 25–33.
21 Strubell E, Verga P, Belanger D, *et al*. Fast and accurate entity recognition with iterated dilated convolutions. Proceedings of 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, Denmark. 2017. 2670–2680.
22 Straková J, Straka M, Hajič J. Neural architectures for nested NER through linearization. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy. 2019. 5326–5331.
23 谢腾, 杨俊安, 刘辉. 基于 BERT-BiLSTM-CRF 模型的中文实体识别. 计算机系统应用, 2020, 29(7): 48–55. [doi: 10.15888/j.cnki.csa.007525]
24 李妮, 关焕梅, 杨飘, 等. 基于 BERT-IDCNN-CRF 的中文命名实体识别方法. 山东大学学报 (理学版), 2020, 55(1): 102–109.
25 李丽双, 郭元凯. 基于 CNN-BLSTM-CRF 模型的生物医学命名实体识别. 中文信息学报, 2018, 32(1): 116–122. [doi: 10.3969/j.issn.1003-0077.2018.01.015]
26 周晓磊, 赵薛蛟, 刘堂亮, 等. 基于 SVM-BiLSTM-CRF 模型的财产纠纷命名实体识别方法. 计算机系统应用, 2019, 28(1): 245–250. [doi: 10.15888/j.cnki.csa.006703]
27 杨文明, 褚伟杰. 在线医疗问答文本的命名实体识别. 计算机系统应用, 2019, 28(2): 8–14. [doi: 10.15888/j.cnki.csa.006760]
28 Guo QP, Qiu XP, Liu PF, *et al*. Star-transformer. Proceedings of 2019 Conference of the North American Chapter of the Association for Computational Linguistics. Minneapolis, MN, USA. 2019. 1315–1325.
29 Kim Y. Convolutional neural networks for sentence classification. Proceedings of 2014 Conference on Empirical Methods in Natural Language Processing. Doha, Qatar. 2014. 1746–1751.
30 Dong CH, Zhang JJ, Zong CQ, *et al*. Character-based LSTM-CRF with radical-level features for Chinese named entity recognition. Proceedings of the 5th CCF Conference on Natural Language Processing and Chinese Computing, NLPCC 2016, and 24th International Conference on Computer Processing of Oriental Languages. Kunming, China. 2016. 239–250.
31 Zhang Y, Yang J. Chinese NER using lattice LSTM. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne, Australia. 2018. 1554–1564.
32 Zhang NX, Li F, Xu GL, *et al*. Chinese NER using dynamic meta-embeddings. IEEE Access, 2019, 7: 64450–64459. [doi: 10.1109/ACCESS.2019.2916816]
33 Zhu YY, Wang GX. CAN-NER: Convolutional attention network for Chinese named entity recognition. Proceedings of 2019 Conference of the North American Chapter of the Association for Computational Linguistics. Minneapolis, MN, USA. 2019. 3384–3393.