Table of Contents

命名实体识别任务的 BERT-BiLSTM-CRF模型

1 命名实体识别任务

输入向量序列：

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_t \cdots, \mathbf{x}_n)$$

其中$\mathbf{x}_t \in \mathbb{R}^m$为t时刻输入向量。
输出序列：

$$\mathbf{y} = (y_1, y_2, \cdots, y_t, \cdots, y_n)$$

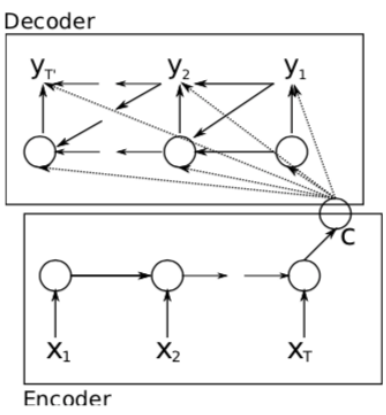
其中$y_t \in \{1, \cdots, k\}$为t时刻输出。

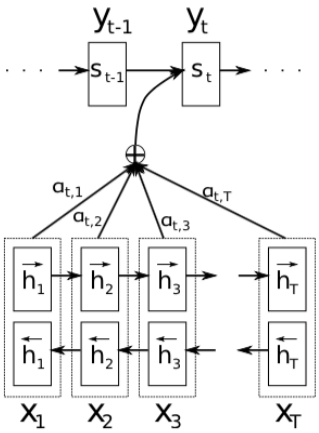
数据样例：
迈/O 向/O 充/O 满/O 希/O 望/O 的/O 新/O 世/O 纪/O 一/O 一/O 一/O 九/O 九/O 八/O 年/O 新/O 年/O 讲/O 话/O （/O 附/O 图/O 片/O 1 /O 张/O ）/O
中/B_nt 共/M_nt 中/M_nt 央/E_nt 总/O 书/O 记/O
国/O 家/O 主/O 席/O 江/B_nr 泽/M_nr 民/E_nr
（/O 一/O 九/O 九/O 七/O 年/O 十/O 二/O 月/O 三/O 十/O 一/O 日/O ）/O
1 /O 2 /O 月/O 3 /O 1 /O 日/O
中/B_nt 共/M_nt 中/M_nt 央/E_nt 总/O 书/O 记/O
国/O 家/O 主/O 席/O 江/B_nr 泽/M_nr 民/E_nr 发/O 表/O 1 /O 9 /O 9 /O 8 /O 年/O 新/O 年/O 讲/O 话/O 《/O 迈/O 向/O 充/O 满/O 希/O 望/O 的/O 新/O 世/O 纪/O 》/O
（/O 新/B_nt 华/M_nt 社/E_nt 记/O 者/O 兰/B_nr 红/M_nr 光/E_nr 摄/O ）/O

人名：nr；地名：ns；组织名：nt。

2 从Encoder/Decoder架构到Attention机制

2.1 Encoder/Decoder架构



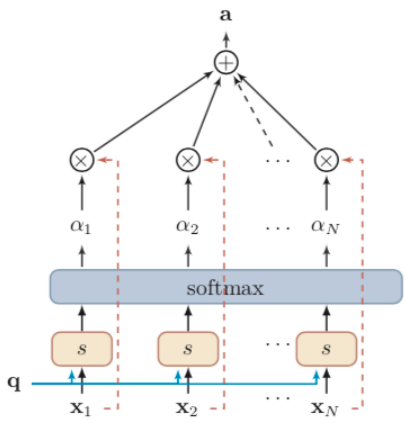


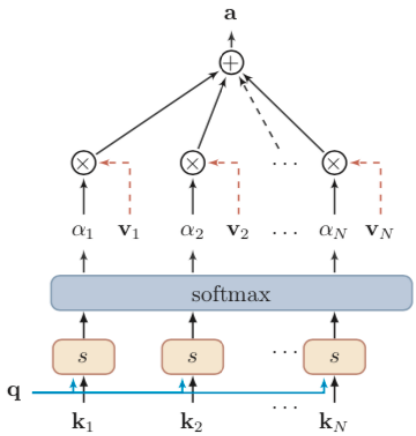
RNN Encoder-Decoder神经网络架构使用循环神经网络学习将变长源序列X编码成定长向量表示$\mathbf{c}$，并将学习的定长向量表示

$\mathbf{c}$解码成变长目标序列$\mathbf{y}$。模型的编码器和解码器被联合训练，以最大化给定源序列的目标序列的条件概率。

seq2seq with Attention神经网络架构中，编码器采用双向循环神经网络学习将输入序列$\mathbf{x}$编码成每个时刻的上下文向量（注意力分布）c_i，解码器学习将上下文向量c_i解码为输出序列$\mathbf{y}$。

2.2 Attention机制





输入序列$X = [\mathbf{x}_1, \cdots, \mathbf{x}_N]$
输出序列$H = [\mathbf{h}_1, \cdots, \mathbf{h}_N]$
注意力机制的计算：

1. 在输入信息上计计算注意力分布；
2. 根据注意力分布计算输入信息的加权平均。

给定一个和任务相关的查询向量$\mathbf{q}$，用注意力变量$z \in [1, N]$表示被选择信息的索引位置，即$z = i$表示选择了第i个输入信息。其中，查询向量$\mathbf{q}$可以是动态生成的，也可以是可学习的参数。

软性注意力的注意力分布
在给定输入信息X和查询变量$\mathbf{q}$下，选择第i个输入信息的概率

$$\begin{aligned}\alpha_i &= p(z = i|X, \mathbf{q}) \\ &= softmax(s(\mathbf{x}_i, \mathbf{q})) \\ &= \frac{\exp(s(\mathbf{x}_i, \mathbf{q}))}{\sum_{j=1}^{N} \exp\left(s\left(\mathbf{x}_j, \mathbf{q}\right)\right)}\end{aligned}$$

其中，α_i称为注意力分布，$s(\mathbf{x}_i, \mathbf{q})$称为注意力打分函数。

注意力打分函数

- 加性模型 $s(\mathbf{x}_i, \mathbf{q}) = \mathbf{v}^{\top} \tanh(W\mathbf{x}_i + U\mathbf{q})$
- 点积模型 $s(\mathbf{x}_i, \mathbf{q}) = \mathbf{x}_i^{\top}\mathbf{q}$
- 缩放点积模型 $s(\mathbf{x}_i, \mathbf{q}) = \frac{\mathbf{x}_i^{\top}\mathbf{q}}{\sqrt{d}}$
- 双线性模型 $s(\mathbf{x}_i, \mathbf{q}) = \mathbf{x}_i^{\top} W\mathbf{q}$

其中，$W, U, \mathbf{v}$为可学习的网络参数，d为输入信息的维度。
加性模型和点积模型的复杂度近似，但点积模型可利用矩阵乘积，计算效率跟高。当输入信息的维度d比较高，点积模型值方差较大，导致softmax函数的梯度较小，缩放点积模型可以解决。双线性模型是泛化的点积模型。若假设$W = U^{\top}V$，则$s(\mathbf{x}_i, \mathbf{q}) = \mathbf{x}_i^{\top}U^{\top}V\mathbf{q} = (U\mathbf{x}_i)^{\top}(V\mathbf{q})$，即分别对$\mathbf{x}_i$和$\mathbf{q}$进行线性变换后进行点积。相比点积模型，双线性模型在计算相似度是引入了非对称性。

注意力函数

$$\begin{aligned}att(X, \mathbf{q}) &= \sum_{i=1}^{N} \alpha_i \mathbf{x}_i \\ &= \mathbb{E}_{z \sim p(z|X,\mathbf{q})}[\mathbf{x}]\end{aligned}$$

2.3 自注意力机制

输入序列$X = [\mathbf{x}_1, \cdots, \mathbf{x}_N] \in \mathbb{R}^{d_1 \times N}$
输出序列$H = [\mathbf{h}_1, \cdots, \mathbf{h}_N] \in \mathbb{R}^{d_2 \times N}$

通过线性变换得到向量序列：

$$Q = W_Q X \in \mathbb{R}^{d_3 \times N}$$
$$K = W_K X \in \mathbb{R}^{d_3 \times N}$$
$$V = W_V X \in \mathbb{R}^{d_2 \times N}$$

其中，Q为查询向量序列，K为键向量序列，V为值向量序列，W_Q, W_K, W_V 分别为可学习参数矩阵。

预测输出向量

$$\begin{aligned}\hat{\mathbf{h}}_i &= att\left((K,V),\mathbf{q}_i\right) \\ &= \sum_{j=1}^{N} \alpha_{ij}\mathbf{v}_j \\ &= \sum_{j=1}^{N} softmax\left(s\left(\mathbf{k}_j,\mathbf{q}_i\right)\right)\mathbf{v}_j\end{aligned}$$

其中，$i, j \in [1, N]$为输出和输入向量序列的位置，连接权重α_{ij}由注意力机制动态生成。

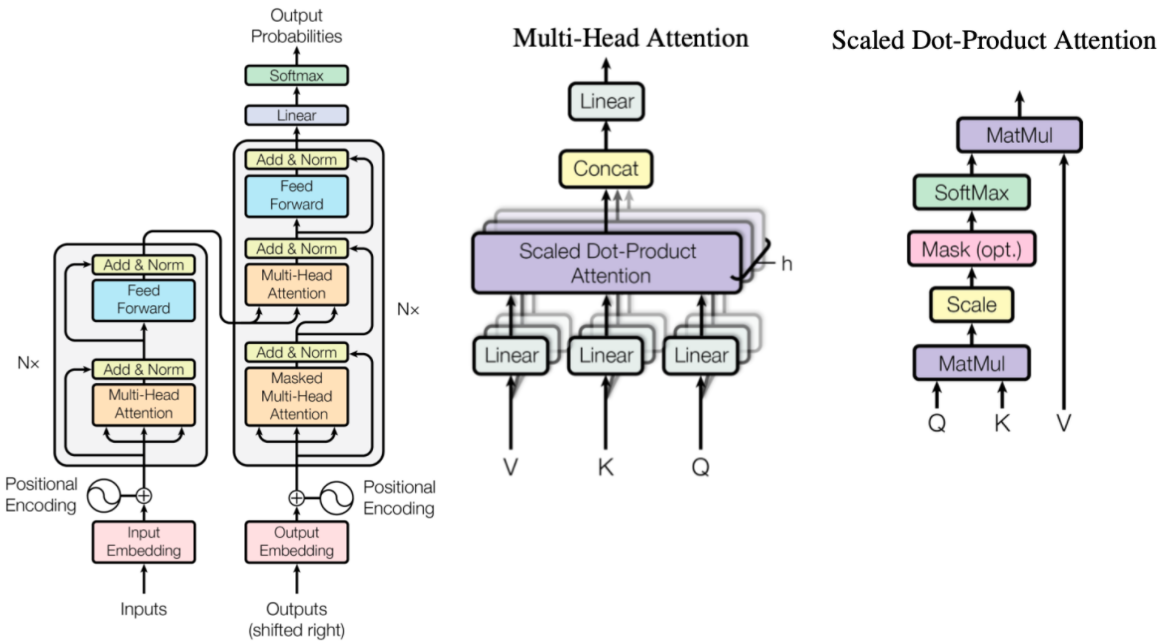
若使用缩放点积模型作为打分函数，则输出向量序列

$$\begin{aligned}H_{d2\times N} &= softmax\left(\frac{K^\top Q}{\sqrt{d_3}}\right)V_{d2\times N} \\ &= softmax\left(\frac{K^\top Q}{\sqrt{d_3}}\right)W_V X\end{aligned}$$

其中，softmax为按列归一化的函数。

3 从Transformer到BERT模型

3.1 Transformer架构



输入序列$inputs = \left(i_1, i_2, \cdots, i_p, \cdots, i_N\right)$，其中$i_p \in \mathbb{N}^*$为输入符号表中的序号。
目标序列$targets = \left(t_1, t_2, \cdots, t_q, \cdots, t_M\right)$，其中$t_q \in \mathbb{N}^*$为目标符号表中的序号。

$$outputs\_probabilities = Transformer\left(inputs, targets\right)$$

其中，$outputs_probabilities = \left(o_1, o_2, \cdots, o_q, \cdots, o_M\right)$为预测序列，$o_q \in \mathbb{N}^*$为目标符号表中的序号。

输入序列词嵌入$Embedding\left(inputs\right) \in \mathbb{R}^{N \times d_{model}}$，其中，$N$为输入序列长度，$d_{model}$为词嵌入维度。

编码器结构：

$$\begin{aligned} e_0 &= Embedding\left(inputs\right) + Pos\_Enc\left(inputs\_position\right) \\ e_l &= EncoderLayer\left(e_{l-1}\right), l \in [1, n] \end{aligned}$$

其中，$e_0 \in \mathbb{R}^{N \times d_{model}}$为编码器输入，$EncoderLayer\left(\cdot\right)$为编码器层，$n$为层数，$e_l \in \mathbb{R}^{N \times d_{model}}$为第$l$层编码器层输出。

编码器层EncoderLayer：

$$\begin{aligned} e_{mid} &= LayerNorm\left(e_{in} + MultiHeadAttention\left(e_{in}\right)\right) \\ e_{out} &= LayerNorm\left(e_{mid} + FFN\left(e_{mid}\right)\right) \end{aligned}$$

其中，$e_{in} \in \mathbb{R}^{N \times d_{model}}$为编码器层输入，$e_{out} \in \mathbb{R}^{N \times d_{model}}$为编码器层输出，$MultiHeadAttention\left(\cdot\right)$为多头注意力机制，$FFN\left(\cdot\right)$为前馈神经网络，$LayerNorm\left(\cdot\right)$为层归一化。

输入向量序列$e_{in} = \left(e_{in1}, e_{in2}, \cdots, e_{inN}\right) \in \mathbb{R}^{N \times d_{model}}$，分别得到查询向量序列$Q = e_{in}$，键向量序列$K = e_{in}$，值向量序列$V = e_{in}$。

多头注意力机制

$$MultiHeadAttention\left(e_{in}\right) = MultiHead\left(Q, K, V\right) = Concat\left(head_1, \cdots, head_h\right) W^O$$

其中，多头输出$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right)$，可学习的参数矩阵$W_i^Q \in \mathbb{R}^{d_{model} \times d_k}, W_i^K \in \mathbb{R}^{d_{model} \times d_k}, W_i^V \in \mathbb{R}^{d_{model} \times d_v}, W^O \in \mathbb{R}^{hd_v \times d_{model}}$

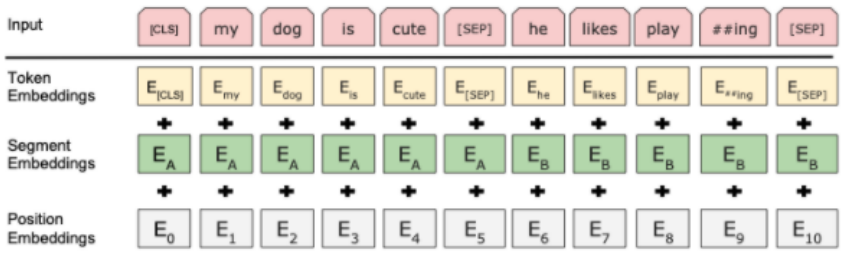
使用缩放点积作为打分函数的自注意力机制

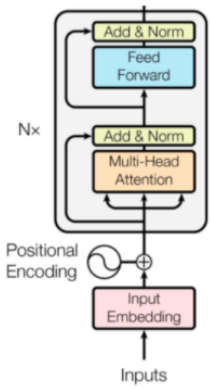
$$Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) = softmax\left(\frac{QW_i^Q\left(KW_i^K\right)^\top}{\sqrt{d_k}}\right)VW_i^V$$

$$\begin{aligned} FFN\left(e_{mid}\right) &= ReLU\left(e_{mid}W_1 + b_1\right)W_2 + b_2 \\ &= \max\left(0, e_{mid}W_1 + b_1\right)W_2 + b_2 \end{aligned}$$

其中，参数矩阵$W_1 \in \mathbb{R}^{d_{model} \times d_{ff}}, W_2 \in \mathbb{R}^{d_{ff} \times d_{model}}$，偏置$b_1 \in \mathbb{R}^{d_{ff}}, b_2 \in \mathbb{R}^{d_{model}}$。

3.2 BERT模型



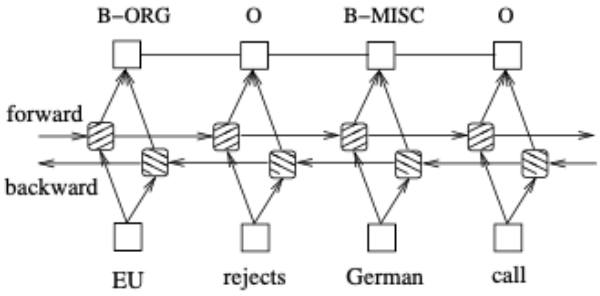


输入序列

$$input = \left([CLS], s_1, s_2, \cdots, s_m, [SEP], p_1, p_2, \cdots, p_n, [SEP]\right)$$

其中，$s_i, p_j \in \mathbb{N}$为输入符号表中的序号；子序列$(s_1, \cdots, s_m)$为句子对中前序句子；子序列$(p_1, \cdots, p_n)$为句子对中后续句子；输入序列首标记$[CLS]$用作分类任务表示；特殊标记$[SEP]$用作区分句子对各子句。

4 BERT-BiLSTM-CRF模型



BiLSTM输出分值矩阵：

$$\mathbf{P} = \left[P_{i,j}\right]_{n\times k}$$

其中$P_{i,j}$为第i个单词对应第j个标签的分数。

转移分值矩阵：

$$\mathbf{A} = \left[A_{i,j}\right]_{(k+2)\times(k+2)}$$

其中$A_{i,j} = p(y_i|y_{i-1})$。

模型分值：

$$s\left(\mathbf{X},\mathbf{y}\right) = \sum_{i=1}^{n+1} A_{i,j} + \sum_{i=1}^{n} P_{i,j}$$

所有可能的标签序列上的softmax产生序列$\mathbf{y}$的概率

$$p(\mathbf{y}|\mathbf{X}) = \frac{e^{s(\mathbf{X},\mathbf{y})}}{\sum_{\tilde{\mathbf{y}}\in\mathbf{Y}} e^{s(\mathbf{X},\tilde{\mathbf{y}})}}$$

预测输出：

$$\begin{aligned}\mathbf{y}^* &= \arg\max_{\tilde{\mathbf{y}}\in\mathbf{Y}} p(\mathbf{y}|\mathbf{X}) \\ &= \arg\max_{\tilde{\mathbf{y}}\in\mathbf{Y}} \log p(\mathbf{y}|\mathbf{X}) \\ &= \arg\max_{\tilde{\mathbf{y}}\in\mathbf{Y}} s(\mathbf{X},\mathbf{y}) \\ &= -\arg\min_{\tilde{\mathbf{y}}\in\mathbf{Y}} s(\mathbf{X},\mathbf{y})\end{aligned}$$

BERT模型生成的Embedding词嵌入表示作为双向LISTM模型的输入。

In [ ]:
```
1
```