文章编号：2095-2163(2020)01-0001-06 中图分类号：TP391 文献标志码：A

序列-序列模型注意力机制模块基本原理探究

马春鹏，赵铁军
（哈尔滨工业大学 计算机科学与技术学院，哈尔滨 150001）

摘 要：编码器-解码器注意力矩阵一直都被认为是传统的神经机器翻译模型（例如基于循环神经网络的模型）学习到的词对齐。然而，通过实验证明了，对于 Transformer 这一结论并不成立。通过比较 Transformer 与基于循环神经网络的模型，研究发现了 2 种模型中注意力机制的本质上的 2 个区别。基于这个观察，提出了 2 种能够让 Transformer 的注意力机制学习到词对齐的方法。实验结果证明了本文提出的方法的有效性，可使 Transformer 既能学习到很好的词对齐，也能够提升机器翻译的性能。

关键词：序列-序列模型；词对齐；Transformer

Research on the fundamental principle of the attention module in sequence-to-sequence model

MA Chunpeng, ZHAO Tiejun
(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

【Abstract】 The encoder-decoder attention matrix has been regarded as the (soft) alignment model for conventional neural machine translation (NMT) models such as RNN-based models. However, it is shown empirically that this is not true for the Transformer. On comparing the Transformer with the RNN-based NMT model, the paper finds two inherent differences, and accordingly presents two methods of capturing word alignments in the Transformer. Experimental results demonstrate the feasibility of the proposed methods. For the Transformer, both the accuracy of the word alignment and the performance of machine translation are improved using the proposed methods.

【Key words】 sequence-to-sequence model; word alignment; Transformer

0 引 言

在基于序列-序列模型的神经网络机器翻译中，编码器和解码器的神经网络结构有很多。常见的结构包括循环神经网络[1-2]、卷积神经网络[3]、自编码神经网络[4]等等。虽然模型的结构有所不同，但是注意力机制模块在各个模型中都存在。

对于机器翻译任务来说，注意力矩阵表示了目标语言句子和源语言句子之间的对应关系。因其与词对齐之间的高度相关性，因此通常被当作是一种概率形式的词对齐模型[5-6]。基于这种思路，有一些研究表明，令词对齐矩阵与真正的词对齐尽量相似，能够提升神经网络机器翻译的性能[7-9]。对于基于卷积神经网络的机器翻译系统来说，词对齐矩阵的可视化输出也表明了其与词对齐之间的相似性（例如，文献[3]的图 3）。

研究又发现，对于自编码神经网络（也被称为 Transformer），注意力矩阵与词对齐之间差异很大。例如，在图 1 中，基于自编码神经网络的模型的注意力矩阵并没有捕捉到英语和汉语单词之间的对应关系，而基于循环神经网络的模型的注意力矩阵与正确的词对齐具有很高的相关性。而且，对于自编码神经网络，这种与词对齐的差异十分普遍。后文会给出关于这一事实的定量分析。

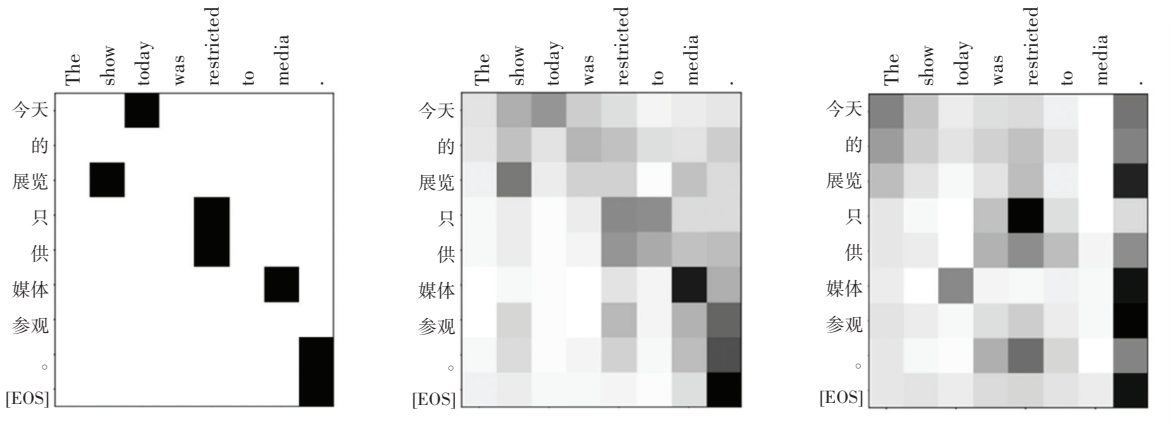
研究观察到的这些现象与之前的关于神经网络机器翻译的研究是矛盾的。之前的研究普遍认为，神经网络机器翻译模型是通过注意力矩阵模块学习词对齐的。因此，为什么基于自编码网络的神经机器翻译模型的注意力矩阵与词对齐有很大的差异，是一个很值得研究的问题。后文将会对这个问题做出解答。实验结果验证了提出的论述，同时，通过向基于自回归网络的神经机器翻译模型中加入若干新的模块，即能使其正确地学习到词对齐。

基金项目：国家重点研发计划项目（2017YFB1002102）。

作者简介：马春鹏（1992-），男，博士研究生，主要研究方向：机器翻译、句法分析、机器学习；赵铁军（1962-），男，博士，教授，博士生导师，主要研究方向：机器翻译、自然语言处理。

通讯作者：赵铁军 Email：tjzhao@hit.edu.cn

收稿日期：2019-08-30



(a) 真正的词对齐矩阵 (b) 基于循环神经网络的模型 (c) Transformer

(a) True word alignment (b) RNN-based model (c) Transformer

图 1 3 个注意力矩阵

Fig. 1 Three attention matrices

1 两种神经网络机器翻译模型的重新表述

为了后文的叙述方便,文中使用同一的数学语言,将 2 种神经网络机器翻译模型(基于循环神经网络的模型与基于自编码神经网络的模型)进行重新表述。对此拟做研究论述如下。

1.1 基于循环神经网络的机器翻译模型

基于循环神经网络的模型在很长一段时间内都是神经网络机器翻译的主流模型,并且已经被部署到了大型的商用系统上[10-11]。通过引入注意力机制模块,机器翻译的性能超过了传统的统计机器翻译方法。

模型由 2 部分组成:编码器和解码器。给定源语言端的句子 $\boldsymbol{w}_s \in \boldsymbol{Z}_+^{n_s}$,其中每一项都表示源语言单词的索引,于是编码器就会生成一个词嵌入的序列。研究将其记作 $\boldsymbol{I}_s^1 \in \boldsymbol{R}^{n_s \times d}$。这里,$n_s$ 是源语言句子的长度,d 是模型的维度。此后,编码器按照下面的方式计算隐含层向量序列 $\boldsymbol{H}_s^1 \in \boldsymbol{R}^{n_s \times d}$。此处需要指出,本文使用方括号表示矩阵的某列,或者向量的某个元素,使用上标和下标来区分不同的矩阵和向量。相应数学公式可表示为:

$$\boldsymbol{H}^1[i] = RNN(\boldsymbol{H}_s^1[i-1], \boldsymbol{I}_s^1[i]), \quad (1)$$

函数 RNN 可以是门循环单元或是长短时记忆网络。下一层的输入可写作如下数学形式:

$$\boldsymbol{I}_s^2 = \boldsymbol{H}_s^1, \quad (2)$$

重复上述计算,就可以得到源语言序列的最终表示 $\boldsymbol{O}_s = \boldsymbol{I}_s^{L_s+1}$,其中 L_s 是编码器的层数。

解码器会按照下面的方式生成目标端的句子 $\boldsymbol{w}_t \in \boldsymbol{Z}_+^{n_t}$,即:

$$\boldsymbol{w}_t[j] = argmax(softmax(FFNN(\boldsymbol{O}_t[j]))), \quad (3)$$

其中,n_t 是目标语言句子的长度。FFNN 是一个前馈神经网络,$\boldsymbol{O}_t = \boldsymbol{I}_t^{L_t+1}$ 是解码器的层数。对于第 l 层,输出 $\boldsymbol{I}_t^{l+1}$ 按照如下方式进行计算,即:

$$\boldsymbol{I}_t^{l+1} = f(\boldsymbol{C}^l, \boldsymbol{H}_t^l), \quad (4)$$

其中,f 是一个非线性函数。$\boldsymbol{H}_t^l$ 按照如下公式进行计算:

$$\boldsymbol{H}_t^l[j] = RNN(\boldsymbol{H}_t^l[j-1], \boldsymbol{I}_t^l[j]), \quad (5)$$

向量 $\boldsymbol{C}^l[j]$ 是 $\boldsymbol{O}_s$ 各个列的加权平均,计算公式具体如下:

$$\boldsymbol{C}^l[j] = \sum_{i=1}^{n_s} softmax((\boldsymbol{H}_t^l[j])^{\mathrm{T}} \boldsymbol{O}_s) \boldsymbol{O}_s[i]. \quad (6)$$

这个被称作是基于循环神经网络的注意力机制。这里只描述了一种被广泛使用的基于循环神经网络的注意力机制,即文献[12]提出的点积注意力机制。

1.2 基于自编码网络的机器翻译模型

与基于循环神经网络的模型相比,基于自编码网络的机器翻译模型最近在速度和精度上都已经超过了前者。这一模型同样使用了序列-序列模型。与基于循环神经网络的模型不同,编码器按照如下的方式计算隐含层的向量,即:

$$\boldsymbol{H}_s^1 = SA(\boldsymbol{I}_s^1, \boldsymbol{I}_s^1[i], \boldsymbol{I}_s^1), \quad (7)$$

其中,$\boldsymbol{I}_s^1$ 不仅包含了词嵌入的信息,还通过位置编码将单词位置的信息编码到了模型中。自注意力机制按照下式进行计算,即:

$$SA(\boldsymbol{K}, \boldsymbol{Q}, \boldsymbol{V}) = softmax\left(\frac{\boldsymbol{K}^{\mathrm{T}}\boldsymbol{Q}}{\sqrt{d}}\right)\boldsymbol{V}, \quad (8)$$

下一层的输入按照如下方式进行计算,即:

$$\boldsymbol{I}_s^2 = FFNN(\boldsymbol{H}_s^1) + \boldsymbol{H}_s^1, \quad (9)$$

这里,研究考虑了网络中的残差连接[13]的情形。对于解码器,$\boldsymbol{I}_t^{l+1}$ 按照下面的方式进行计算,即:

$$\boldsymbol{I}_t^{l+1} = FFNN(\boldsymbol{H}_t^l) + \boldsymbol{H}_t^l, \quad (10)$$

$$\boldsymbol{H}_t^l[j] = SA(\boldsymbol{O}_s, \widetilde{\boldsymbol{H}}_t^l[j], \boldsymbol{O}_s), \quad (11)$$

$$\widetilde{\boldsymbol{H}}_t^l[j] = SA(\boldsymbol{I}_t^l, \boldsymbol{I}_t^l[j], \boldsymbol{I}_t^l). \quad (12)$$

上面的方程中的自注意力机制就是本论文研究的主题。

2　循环神经网络注意力机制与自编码网络注意力机制的比较

文中猜测，对于基于自编码网络的机器翻译系统，注意力矩阵与词对齐并不相关。为了验证这一猜测，研究通过实验比较了 2 种神经网络机器翻译模型在词对齐任务上的效果。

2.1　实验配置与基线系统

文中使用 LDC 数据集来训练英语-汉语的神经网络机器翻译模型。LDC 语料库由以下部分构成：LDC2002E18、LDC2003E07、LDC2003E14、LDC2004T07 的 Hansards 部分、LDC2004T08 以及 LDC2005T06。合计约 140 万平行句对。翻译性能根据单词粒度的 *BLEU* 得分[14]进行评价。选择使用 NIST MT 2002 数据集进行评价。这个数据集含有 878 个平行句对。

由于在 NIST MT 2002 中，没有人工标记的词对齐信息，因此使用一个人工标注的词对齐语料库（THU 语料库，http://nlp.csai.tsinghua.edu.cn/~ly/systems/TsinghuaAligner/TsinghuaAligner.html）来评价词对齐的学习质量。这个语料库由英语-汉语平行句对组成，这些句对的词对齐信息已经被人工标注完毕。每个词对齐信息都关联着一个标注人员的确信程度（“确信”或“不确信”）。研究将包含“不确信”的句对全部删除。为了提升评价的可信性，只评价长句子（即，包含 10 个词对齐以上的句子）。最终的 THU 语料库包含 854 个平行句对。需要注意的是，虽然 THU 语料库还提供了 130 万的平行句对用于训练一个词对齐模型，但是只使用了 THU 语料库的测试集部分。评价的度量是词对齐错误率（*AER*）。在评价 *AER* 时，研究强制令解码器输出参考译文的单词，选择词对齐矩阵中的最大值作为对齐的源语言单词，进行评价。

2 种神经网络机器翻译模型都是基于 OpenNMT（http://opennmt.net）[15]实现的。对于基于循环神经网络的机器翻译模型，编码器和解码器都有 2 个隐含层，隐含层的单元是长短时记忆网络。对于自编码网络的神经机器翻译模型，编码器和解码器的层数均为 6。研究使用了多头注意力机制，头的数量为 8。同时还使用了层归一化策略[16]。关于模型的正则化，则使用了下面的方法：标签平滑[17]和 dropout[18]。在优化时，选择使用了 Adam 优化算法[19]。

表 1 给出了基线系统的实验结果。对于基于自编码神经网络的机器翻译模型，由于采用了多头注意力机制，词对齐是通过最后一个头进行计算的。虽然自编码网络的机器翻译模型的翻译质量要远好于另一方，但是注意力矩阵给出的词对齐的质量要远差于另一方。

表 1　基线系统的实验结果

Tab. 1　Experiment results of the baseline system

系统	*AER*	*BLEU*（MT02）	*BLEU*（THU）
RNN-NMT	18.9	22.11	11.21
Transformer	28.2	25.50	28.18

2.2　自注意力机制不同头的效果

之前已经有研究表明，对于多头自注意力机制来说，调节头的数量[20]或者对各个头取平均[21]会对模型的性能产生很大的影响。因此，研究考察了在学习词对齐的任务上，调节自注意力机制的头会产生怎样的影响。

表 2 给出了自注意力机制的不同头计算得到的词对齐错误率。由表 2 可以看到，虽然词对齐错误率各不相同，但是所有的头都没有很好地学习到词对齐。所有的词对齐错误率都要远高于基于循环神经网络的机器翻译模型的注意力模块计算得到的词对齐错误率（18.9）。

表 2　Transformer 不同头的 *AER*

Tab. 2　*AERs* of different heads of the transformer

头编号	0	1	2	3	4	5	6	7
AER	31.6	31.8	31.9	30.7	30.1	34.6	29.2	28.2

表 3 给出了调节自注意力机制头的数量的结果，以及对各个头取平均的结果。表 3 中，井号（#）表示头的数量，“h_{last}”表示使用最后一个头计算词对齐错误率，“*aver*”表示使用所有头的平均值来计算词对齐错误率。

表 3　调节自注意力机制头数目的效果

Tab. 3　Effects of modifying the number of heads of self-attention mechanism

系统	*AER*	*BLEU*（MT02）	*BLEU*（THU）
#=1, h_{last}	31.3	23.35	28.30
#=2, h_{last}	27.9	23.87	27.82
#=4, h_{last}	30.1	24.76	28.03
#=8, h_{last}	28.2	25.50	28.18
#=16, h_{last}	29.1	25.24	28.55
#=8, *aver*	32.3	25.50	28.18

可以看到,虽然机器翻译的性能几乎会随着头数量的增加而变好,但是词对齐错误率几乎不变。对所有头取平均也不会让词对齐错误率有所降低。因此,对于基于自编码网络的神经机器翻译模型来说,仅仅调节头的数目是不够的,并不能够让模型学习到很好的词对齐。

2.3 训练阶段模型的演化

图 2 给出了单词粒度的 *BLEU* 得分与词对齐错误率在训练过程中的变化情况。*BLEU* 得分是在 NIST MT 02 语料库上测试得到的,词对齐错误率是在 THU 语料库上测试得到的。正如研究前期预想的那样,2 个模型的 *BLEU* 得分都会随着训练的进行而逐渐升高,并且基于自编码网络的机器翻译模型会得到更好的翻译效果。然而,自编码网络的神经机器翻译模型的词对齐错误率要比基于循环神经网络的模型更高,并且会随着训练的进行而变得越来越高,也就是说词对齐的效果会越来越差。这就为前文的猜想提供了一个证据,即,基于自编码神经网络的神经机器翻译系统的注意力矩阵并不是词对齐。

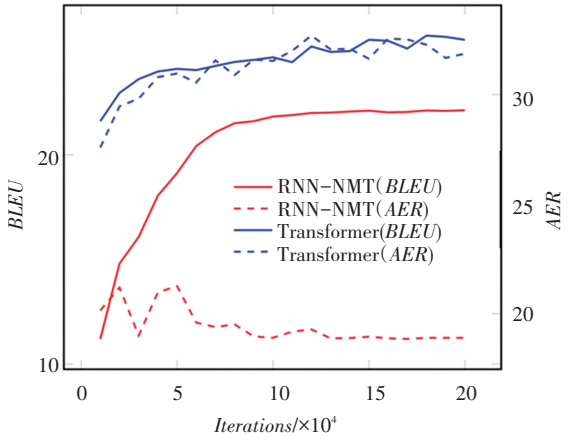


图 2 *AER* 与单词粒度 *BLEU* 得分的演化过程

Fig. 2 Evolution of *AER* and word-level *BLEU* score

2.4 有监督注意力机制方法的效果

遵循文献[8]的做法,研究使用金标准的词对齐来引导模型的训练。对于基于自编码网络的神经机器翻译系统,只对多头自注意力机制的最后一个头进行引导。具体地,首先将金标准的词对齐转化为 0-1 的矩阵,再使用一个服从正态分布 $N(0,0.5)$ 的高斯滤波器对矩阵进行平滑。然后,在训练时,将在损失函数中加入下面一项。具体如下:

$$d(\boldsymbol{A},\boldsymbol{A}^{*}) = \left(\sum_{t=1}^{n_t}\sum_{s=1}^{n_s}(\boldsymbol{A}^{*}[t,s]-\boldsymbol{A}[t,s])^{2}\right)^{\frac{1}{2}}. \tag{13}$$

其中,$\boldsymbol{A}^{*}$ 是平滑后的金标准词对齐矩阵,$\boldsymbol{A}$ 是神经网络机器翻译模型学习得到的注意力矩阵。

对于文中的实验,LDC 训练语料库的金标准词对齐矩阵是使用 GIZA++(http://www.fjoch.com/GIZA++.html)工具得到的。表 4 给出了有监督注意力机制方法的实验结果。对于基于循环神经网络的机器翻译模型,使用有监督注意力机制的方法,词对齐的错误率有所降低,机器翻译的性能有所提升。然而,对于基于自编码神经网络的机器翻译模型,使用有监督注意力机制的方法,虽然词对齐的错误率得到了大幅度的降低,但是机器翻译的性能受到了很大程度的损害。这就证明了本次研究中的假设:自编码神经网络的机器翻译模型的注意力矩阵与词对齐是有很大差异的,因此金标准的词对齐会误导训练过程的进行。

表 4 有监督注意力机制的实验结果

Tab. 4 Experiment result of supervised attention mechanism

系统	*AER*	*BLEU*(MT02)	*BLEU*(THU)
RNN-NMT(基线)	18.9	22.11	11.21
RNN-NMT(有监督)	17.1	22.21	11.72
Transformer(基线)	28.2	22.50	28.18
Transformer(有监督)	21.5	23.52	25.67
GIZA++	19.3	-	-

3 自注意力机制与词对齐不匹配的原因

通过上述实验分析,一个很自然的问题就是:为什么自注意力机制与词对齐不存在对应关系。这就是本节所关注的问题。

通过比较基于循环神经网络的序列-序列模型的注意力机制与基于自编码网络的序列-序列模型的注意力机制,可以看到,除了一个常数 $\sqrt{d}$,两者之间几乎完全相同。因此,则会猜想,这个问题的答案不在于注意力机制本身,而在于编码器隐含层表示 $\boldsymbol{H}_s^l$ 和解码器隐含层表示 $\boldsymbol{H}_t^l$ 的计算方法上。同时还发现,在计算隐含层表示时,有 2 个主要的不同:暴露范围与依赖关系。故而,研究有针对性提出了 2 种学习更优质的词对齐的方法,并验证了这 2 种方法的有效性。

3.1 暴露范围的不同

当计算编码器的表示时,除了表面上使用的具体数学公式有所不同外,研究发现,其根本性的不同在于暴露范围的不同。

对于基于循环神经网络的机器翻译模型,$\boldsymbol{H}_s^l[i]$ 是通过循环神经网络递归地计算得到的。参

与计算的变量包括了所有的 $\boldsymbol{H}_s^l[k]$。对于前向循环神经网络，$k \in [1, i-1]$，而对于后向循环神经网络，$k \in [i+1, n_s]$。虽然 2 个方向的计算结果最后要被合并到一个向量中，但是对于最后的隐含层向量表示中的某一个具体的元素来说，只有某一些 $\boldsymbol{H}_s^l[k]$ 参与了计算。

相反，对于基于自编码网络的机器翻译模型来说，编码器的隐含层向量表示中的任何一个元素都使用了所有的 $\boldsymbol{H}_s^l[k]$ ($k \in [1, n_s]$) 来进行计算。虽然这些多出来的信息或许能够提升机器翻译的性能，但却会误导词对齐的学习。

考虑了这一原因后，故而猜想，在基于自编码神经网络的机器翻译模型的编码器一侧加入一个编码遮罩或许能够使其学习到质量更好的词对齐。具体地，在计算 $\boldsymbol{H}_s^1[i]$ 时，将按照下面的方式进行计算，即：

$$\boldsymbol{H}_s^1 = SA(\boldsymbol{I}_s^1[\leqslant i], \boldsymbol{I}_s^1[i], \boldsymbol{I}_s^1[\leqslant i]), \quad (14)$$

研究将这种计算方式称作前向编码遮罩。也可以按照下面的方式计算 $\boldsymbol{H}_s^1[i]$，即：

$$\boldsymbol{H}_s^1 = SA(\boldsymbol{I}_s^1[\geqslant i], \boldsymbol{I}_s^1[i], \boldsymbol{I}_s^1[\geqslant i]). \quad (15)$$

这种计算方式被称作后向编码遮罩。编码器的其它层也可以按照类似的方式进行计算。

表 5 给出了添加编码遮罩的实验结果。星号表示实验结果具有统计显著性。由表 5 可以看到，虽然词对齐的错误率仍然要高于基于循环神经网络的机器翻译模型，但是通过添加编码遮罩的方式，确实能够让基于自编码网络的机器翻译模型学习到更好的词对齐。此外，虽然编码遮罩减少了编码器所使用的信息，但是在 THU 语料库上的翻译效果并没有受到太大的影响。对于 NIST MT 02 语料库，*BLEU* 得分在一定程度上甚至还有所上升，这一点就超出了设计预期。因此，分析后可知，暴露范围的不同确实是造成自编码网络的神经机器翻译模型无法成功学习到词对齐的原因之一。

表 5　加入编码遮罩的效果

Tab. 5　Effects of adding encoding masks

系统	*AER*	*BLEU* (MT02)	*BLEU* (THU)
基线(Transformer)	28.2	25.50	28.18
+前向编码遮罩	25.1	26.66	27.49
+后向编码遮罩	25.4	25.92	28.08

3.2　依赖关系的不同

在计算解码器的隐含层表示时，研究发现，对于基于循环神经网络的模型，$\boldsymbol{H}_t^l[j]$ 依赖于 $\boldsymbol{H}_t^l[j-1]$（即，同一层的历史状态）。然而，对于基于自编码网络的模型，$\boldsymbol{H}_t^l[j-1]$ 没有参与计算。$\boldsymbol{H}_t^l[j]$ 依赖于 $\boldsymbol{I}_t^l$，从而进一步依赖于 $\boldsymbol{H}_t^{l-1}$（即，前一层的全部状态）。对于编码器，这一结论也成立。因此有理由相信，对于学习词对齐来说，同一层的历史状态是很有用的。

然而，由于自编码网络的并行处理的特性，将同一层的历史信息加入到隐含层向量的计算中并不容易。研究提出了一种折中的解决方案。在基于自编码神经网络的机器翻译模型的编码器或解码器的最后一层的上面，加上一层循环神经网络。具体地，$\widetilde{\boldsymbol{O}}_s$ 按照如下方式进行计算，即：

$$\widetilde{\boldsymbol{O}}_s = PFN(LSTM(FFNN(\boldsymbol{O}_s) + \boldsymbol{O}_s)). \quad (16)$$

其中，PFN 是按照位置的前馈神经网络。类似地，解码器的最后一层的隐含层表示 $\widetilde{\boldsymbol{O}}_t$ 也可以得到。

表 6 给出了在编码器或解码器的最后一层上方添加循环神经网络层的效果。研究选择的循环神经网络是双向的长短时记忆单元网络。表 6 中的井号(#)表示循环神经网络的层数，星号表示实验结果具有统计显著性。

表 6　添加双向循环神经网络的结果

Tab. 6　Experiment results of adding bidirectional RNNs

LSTM	*AER*	*BLEU* (MT02)	*BLEU* (THU)
编码器(#=1)	29.3	25.12	28.32
编码器(#=2)	25.6	25.88	27.86
解码器(#=1)	28.6	25.76	28.33
解码器(#=2)	25.2	26.33	28.00
编码器+解码器(#=2)	25.3	26.10	28.43
基线	28.2	25.50	28.18

从实验结果中，可以看到，在编码器或解码器一侧添加循环神经网络并没有显著的差别。随着循环神经网络层数的增加，模型可以学习到更好的词对齐。而且，在大多数情形下，机器翻译的效果都有所提升。这个实验就证明了，依赖关系的不同也是造成基于自编码网络的机器翻译系统无法学习到优质词对齐的原因之一。

4　结束语

研究发现 Transformer 的注意力机制矩阵并不对应于词对齐。研究设计了多组实验，通过实验数据，定量地证明了这一点。同时，分析给出了这个现象的原因，并且提出了 2 种方法，使其能够成功地学习到词对齐。

参考文献

[1] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks[C]//Proceedings of Advances in Neural Information Processing. Montréal, Quebec, Canada: MIT Press, 2014: 3104-3112.

[2] CHEN Miaxu, FIRAT O, BANPA A, et al. The best of both worlds: Combining recent advances in neural machine translation [J]. arXiv preprint arXiv: 1804.09849, 2018.

[3] GEHRING J, AULI M, GRANGIER D, et al. Convolutional sequence to sequence learning [C]//Proceedings of the 34th International Conference on Machine Learning. Sydney, NSW, Australia, :dblp, 2017: 1243-1252.

[4] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [C]//31st Conference on Neural Information and Processing Systems. Long Beach, CA, USA: dblp, 2017: 5998-6008.

[5] KNOWLES R, KOEHN P. Context and copying in neural machine translation[C]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: ACL, 2018: 3034-3041.

[6] SHANKAR S, GARG S, SARAWAGI S. Surprisingly easy hard-attention for sequence-to-sequence learning [C]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: ACL, 2018: 640-645.

[7] LIU Lemao, UTIYAMA M, FINCH A, et al. Neural machine translation with supervised attention [C]//Proceedings of COILING 2016. the 26th International Conference on Computational Linguistics. Osaka, Japan: [s. n.], 2016: 3093-3102.

[8] MI Haitao, WANG Zhiguo, ITTYCHERIAH A. Supervised attention for neural machine translation[J]. arXiv preprint arXiv: 1608.00112, 2016.

[9] DEVLIN J, CHANG Mingwei, LEE K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding [C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis, Minnesota, USA : ACL, 2019: 4171-4186.

[10] WU Yonghui, SCHUSTER M, CHEN Zhifeng, et al. Google's neural machine translation system: Bridging the gap between human and machine translation[J]. arXiv preprint arXiv: 1609.08144, 2016.

[11] CREGO J, KIM J, KLEIN G, et al. Systran's pure neural machine translation systems [J]. arXiv preprint arXiv: 1610.05540, 2016.

[12] LUONG M T, PHAM H, MANNING C D. Effective approaches to attention-based neural machine translation [J]. arXiv preprint arXiv: 1508.04025, 2015.

[13] HE Kaiming, ZHANG Xiangyu, REN Shaoqing, et al. Deep residual learning for image recognition [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA: IEEE, 2016: 770-778.

[14] PAPINENI K, ROUKOS S, WARD T, et al. BLEU: A method for automatic evaluation of machine translation[C]//Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Morristown, NJ, USA: ACL, 2002: 311-318.

[15] KLEIN G, KIM Y, DENG Yutian, et al. OpenNMT: Open-source toolkit for neural machine translation[C]//Proceedings of ACL 2017, System Demonstrations. Vancouver, Canada: ACL, 2017: 67-72.

[16] BA J L, KIROS J R, HINTON G E. Layer normalization[J]. arXiv preprint arXiv: 1607.06450, 2016.

[17] SZEGEDY C, VANHOUCKE V, IOFFE S, et al. Rethinking the inception architecture for computer vision[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA: IEEE, 2016: 2818-2826.

[18] SRIVASTAVA N, HINTON G, KRIZHEVSKY A, et al. Dropout: A simple way to prevent neural networks from overfitting[J]. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

[19] KINGMA D, BA J. Adam: A method for stochastic optimization [J]. arXiv preprint arXiv: 1412.6980, 2014.

[20] TANG Gongbo, MULLER M, RIOS A, et al. Why self-attention? A targeted evaluation of neural machine translation architectures [C]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: dblp, 2018: 4263-4272.

[21] ZENKEL T, WUEBKER J, DENERO J. Adding interpretable attention to neural translation models improves word alignment [J]. arXiv preprint arXiv: 1901.11359, 2019.