基于掩码机制的非自回归神经机器翻译

贾浩，王煦，季佰军，段湘煜，张民

（苏州大学计算机科学与技术学院，江苏 苏州，215000）

摘要:当前基于自注意力机制的神经机器翻译模型取得了长足的进展，但是采用自回归的神经机器翻译在解码过程中无法并行计算，耗费时间过长。我们提出了一个采用非自回归的神经机器翻译模型，可以实现并行解码，并且只使用一个 Transformer 的编码器模块进行训练，简化了传统的编码器-解码器结构。同时在训练过程中我们引入了掩码机制，减小了与自回归的神经机器翻译的翻译效果差距。相比其他非自回归翻译模型，在 WMT2016 罗马尼亚语-英语翻译任务上我们取得了更好的效果，并且使用跨语言预训练语言模型初始化之后，我们取得了和自回归神经机器翻译模型相当的结果。

关键词：神经机器翻译；掩码机制；非自回归

机器翻译[1] [2]的主要研究目的是通过计算机实现将一种语言自动翻译成另一种语言，传统的统计机器翻译(statistic machine translation ,SMT)[3]系统性能提升缓慢，但是近年来，随着神经机器翻译(neural machine translation ,NMT)[4]的提出，机器翻译性能大幅度提升，再次得到了广泛的关注。

NMT 模型一般采用编码器-解码器[5]结构，使用编码器学习源端单词的上下文表征，之后再通过解码器来预测目标端的单词。在 2014 年 Bahdanau 等人[6]第一次将注意力机制引入 NLP 领域，进一步提升了 NMT 的性能。之后 Luong 等人[7]拓展了注意力机制在机器翻译的领域运用，提出了全局注意力与局部注意力的概念。在 2017 年，Vaswani 等人[8]提出了完全基于注意力机制的翻译模型 Transformer，它通过自注意力机制（Self-attention）将输入序列中不同位置的信息相互联系起来，并且与之前的递归结构、卷积结构相比复杂度更低，速度更快，同时最终的翻译效果也有较大提升。

虽然在翻译效果上取得了较大进步，但相关模型也越来越复杂，并且编码器与解码器分开设计，进一步增加了模型的复杂度。为了简化传统的机器翻译模型框架，我们提出了基于掩码机制的非自回归神经机器翻译模型。此模型只使用一个带有自注意力机制的编码器，并且运用了类似 Devlin 等人[9]在其 MLM（Masked Language Model，掩码语言模型）中使用的掩码机制。在训练过程中，我们使用类似于 Lample 等人[10]的方法，将平行源端句子与目标端句子拼接输入，但是我们只对目标端的单词进行掩码处理。在训练过程中模型需要依靠自注意力层学习源端句子和目标端未被掩码处理的句子信息，来预测目标端被掩码处理的单词。并且通过这个模型，我们可以同时实现非自回归的神经机器翻译，与传统的自回归的神经机器翻译相比，可以通过并行计算大大提高翻译的速度。

本文通过使用掩码机制的解码器模块，简化了传统的编码器-解码器结构，并且可以实现非自回归的神经机器翻译，在 WMT 2016 罗马尼亚语-英语翻译任务中相比与其他的非自回归机器翻译模型取得了更好的结果，同时，用跨语言预训练语言模型初始化之后，我们取得了和自回归神经机器翻译模型相当的结果。

1 相关工作

目前已经有一些工作尝试缩小编码器与解码器之间的差异，将它们简化为一个模块。Bapna 等人[11]尝试将编码器所有层的信息传给解码器。而 He 等人[12] 将编码器与解

码器参数共享，取得了不错的效果，而Fonollosa 等人[13]则在其基础上对注意力层添加了局部约束。这些方法都尝试将编码器与解码器在注意力层面上统一，但是在编码过程中之前的模型只能依靠源端句子，而我们的模型会同时考虑源端句子和部分目标端句子，因此能更好地生成目标句子。并且在最后的解码过程中，之前的模型只能使用从左到右的自回归解码方式，而我们的模型则能采用非自回归解码方式，大大提高了解码速度。

非自回归神经机器翻译近几年在神经机器翻译领域逐渐受到关注，传统的自回归神经机器翻译在解码过程中每次以所有先前单词为条件预测下一个单词，而非自回归神经机器翻译则并行的一次生成所有预测单词，虽然在配置相同的情况下非自回归神经机器翻译效果往往没有自回归神经机器翻译好，但其翻译速度相对有大幅提高。Gu等人[14]首先提出了非自回归神经机器翻译，与传统 Transformer 模型相比，多了一个Fertility Predictor 模块，使用外部的对齐工具(Fast align)来生成 fertility 信息，用来决定输出句子长度，之后再并行地生成目标语句的各个单词。Lee 等人[15] 则通过编码器隐状态直接生成句子长度，并且采用多次迭代的方法，直接取上一轮翻译的结果作为下一轮的解码器输入，从而提升翻译效果。Ghazvininejad 等人[16]则在 Lee 等人[15]的基础上采用掩码机制，随机对解码器的输入进行掩码处理，而输出则为这些被掩码的单词。为了得到更合适的翻译结果，Wei 等人[17]将自回归模型作为教师模型，指导模型的训练。Shao等人[18]则引入了强化学习的训练方法。但是，这些方法仍然需要使用编码器-解码器结构，在我们的工作中，我们使用一个统一的模块建立了非自回归神经机器翻译模型，在编码与解码过程中，模型都能灵活地使用源端和目标端的信息。

我们在训练过程中运用了类似 Devlin等人[9]在其 MLM 中使用的掩码机制，这个方法最先被用于单语言预训练中，之后Lample 等人[19] 将其运用到跨语言预训练中，他们将源端语句和目标端语句拼接作为输入，并且在源端语句和目标端语句中都随机掩盖一部分词进行训练，最终的训练目标是预测出被随机掩盖的单词。同时Ghazvininejad 等人[20]通过对句子不同位置进行掩码处理，经过多轮重复预测句子，提高最终的预测效果。在本文中，我们也采用相似的方法，但是我们只对目标端语句进行掩码处理，同时在对源端信息进行注意力运算时，遮掩掉目标端信息，训练模型预测出相应的单词以及预测出目标端句子的长度。

2 模型结构

2.1 结构基础

受掩码语言模型 MLM[9]的启发，我们采用带有双向自注意力机制的 Transformer的编码器作为我们实验的基本模型结构，如图 1 所示。模型由一个 Transformer 编码器和一个单层 RNN 组成，RNN 模块用来预测目标语言的句子长度，Transformer 编码器用来预测目标语言的句子内容。这样一来，相比于传统的自回归神经机器翻译，我们的模型能减少近一半的参数，大大减少了模型的参数规模。

Transformer 编码器由六个编码器层堆叠而成，每个编码器层包含两个子层：多头自注意力子层和前馈神经网络子层，每个子层之后都要经过残差连接和层标准化。

我们提出的模型框架融合了掩码机制、非自回归神经机器翻译、遮掩-预测解码机制等思想。

2.2 掩码机制

MLM 实质上是一个带有双向自注意力机制以及掩码机制的 Transformer 的编码器，在此结构基础上，模型能够在每一层中联合上下文以学习到双向表示。因此，我们采用基于 MLM 的训练目标来优化输入文本的上下文表示。具体来说，我们将输入文本中的词随机替换为特殊标记[*mask*]，实现对输入文本的部分遮掩，然后模型通过学习文本的上下文信息来预测出被遮掩的词。

而在我们的模型中，采用了一个带有

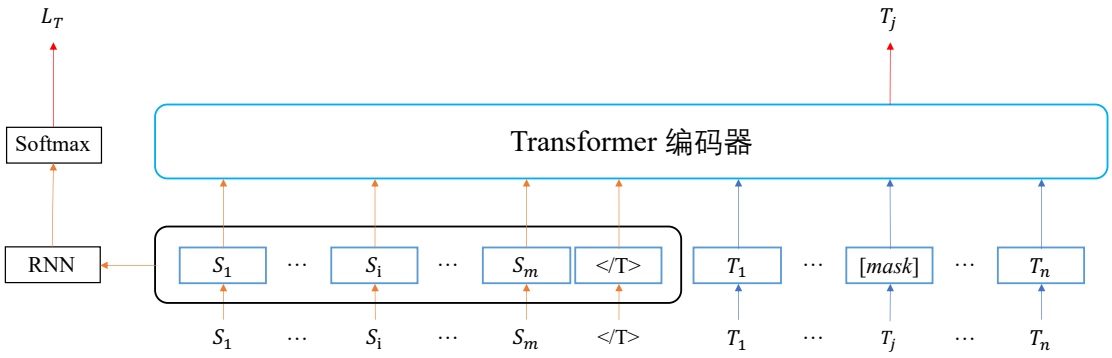

图 1 模型结构

Fig.1 Structure diagram of our model

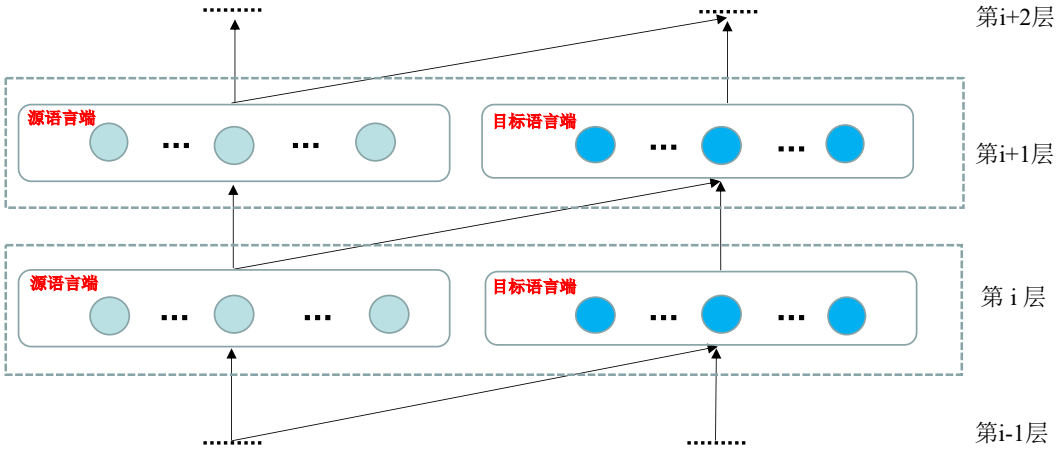

图 2 每个编码器层之间的注意力计算方式

Fig.2 Attention calculation method between each encoder layer

掩码机制的 Transformer 编码器。在模型训练时，我们对目标语言句子进行随机的遮掩之后，将双语平行句对用[/T]隔开，作为一个序列，进入 Transformer 编码器进行自注意力的计算。其中，如图 2 所示，对于源语言句子部分，在进行自注意力运算时，将目标语言句子部分的权重置为负无穷，即让模型在进行源语言部分注意力计算时，感知不到目标语言的句子；而在进行目标语言句子部分的自注意力运算时，不对源语言句子部分的权重进行任何的操作，即让模型在目标语言部分注意力计算时，能够感知到整个句对。

2.3 非自回归神经机器翻译

自回归神经机器翻译在对目标语言的每个词进行预测时，依赖于预测词前面位置的词的信息，因此，只能一个词一个词地进行预测；而非自回归神经机器翻译对目标语言的每个词独立地进行预测，不依赖于预测词前面位置的单词，因此能一次性地预测出整个目标语言句子。

给定一个源语言句子 $s_1^I = s_1, \dots, s_i, \dots, s_I$，我们模型的目标 p_θ 是生成正确的目标语言句子 $t_1^J = t_1, \dots, t_j, \dots, t_J$：

$$p(t_1^J|s_1^J) = \prod_{j=1}^{J} p(t_j|s_1^I, t_1^{j-1}, t_{j+1}^J) = \prod_{j=1}^{J} p_\theta(t_j|s_1^I, \tilde{T}_j) \quad (1)$$

其中，$\tilde{T}_j = \tilde{T}(t_1^{j-1}, t_{j+1}^J)$，即对于目标语言句子$t_1^J$随机遮掩掉的单词$t_j$，其前后文信息。因此，只要给定源语言句子和经过随机掩码的目标语言句子，通过我们的模型就可以对目标语言句子的每个位置独立地进行建模，不依赖于其前面位置的输出。

我们用N表示被遮掩掉的单词个数，模型的遮掩、生成部分的损失函数可表示为：

$$Loss1 = -\frac{1}{N}\sum_{i=1}^{N} log\, P_\theta(t_i|s_1^I, \tilde{T}_j) \quad (2)$$

2.4 长度预测模块

在传统的自回归神经机器翻译中，从左往右逐词解码，直到预测出代表句子结束的特殊标识符</s>，作为句子解码结束的标志。而在非自回归神经机器翻译中，同时预测出整个句子的内容，所以我们在预测句子内容之前，需要提前知道句子长度。因此，我们采用了一个单层的循环神经网络（Recurrent Neural Network, RNN），来提前预测句子的长度。

假设目标语言句子的最大长度为L_{max}，对于源语言句子的输入$s_1, \dots, s_i, \dots, s_I$，我们可以通过如下公式依次得到隐状态$H_1, \dots, H_i, \dots, H_I$：

$$H_t = tanh(WH_{t-1} + Us_t), t = 1,2, \dots, I \quad (3)$$

然后根据 I 位置的隐状态H_I，我们通过如下公式做一个 softmax 线性变换，将H_I映射为L_{max}维的向量，把长度预测问题转化为一个多分类问题。

$$\hat{y}_I = softmax(VH_I) \quad (4)$$

其中，W，U，V都为矩阵。对于$\hat{y}$，我们得到概率值最大的维度L，即为我们的预测出的目标端句子的长度。

因此，在模型训练阶段，我们将目标端句子的真实长度编码成一个L_{max}维的独热向量（one-hot vector）Y，长度预测模块的训练目标为最小化交叉熵（Cross Entropy，CE）损失：

$$Loss2 = CE(Y, \hat{y}_I) \quad (5)$$

2.5 损失函数

结合长度预测模块和目标端句子预测模型，我们模型的损失函数由两部分组成。在模型训练过程中，最小化损失函数：

$$Loss = Loss1 + Loss2 \quad (6)$$

2.6 遮掩-预测解码机制

自回归神经机器翻译通常是从左到右逐词解码，而非自回归神经机器翻译由于一次性解码目标端的所有词，虽然能够提高解码的速度，但是如何提高解码的效果至关重要。因此，在模型解码阶段，我们采用“遮掩-预测”解码机制[16]（Mask-Predict）进行逐步优化，提高模型解码出的句子效果。

解码的算法思路如下：首先我们根据源语言的句子，通过长度预测模块得到目标语言句子的长度 L，然后将目标语言句子置为 L 个特殊标记[$mask$]，与源语言句子拼接在一起进入模型，预测出目标语言句子中的所有词。之后进入 N 轮的迭代优化，每一轮的优化迭代包含遮掩和预测两部分。

遮掩：在第 t 轮，对于上一轮预测得到的目标端句子，遮掩掉句子中预测概率p_i最低的 k 个词$T_{mask}^{(t)}$后，得到目标端句子$\widetilde{T^{(t)}}$，

其中，

$$k = L \cdot \frac{N-t}{N} \quad (7)$$

$$T_{mask}^{(t)} = \operatorname{argmin}_i(p_i, k) \quad (8)$$

预测：将源语言句子s_1^l和遮掩之后的目标端语言句子$\widetilde{T^{(t)}}$拼接在一起进入模型，预测出被遮掩掉的词$y_i^{(t)}$及其预测概率，并对这部分词的预测概率进行更新，形成新的目标端句子$\widehat{T^{(t)}}$。

$$y_i^{(t)} = \operatorname*{argmax}_w P\left(y_i = w \middle| s_1^l, \widetilde{T^{(t)}}\right) \quad (9)$$

$$p_i^{(t)} = \max_w P\left(y_i = w \middle| s_1^l, \widetilde{T^{(t)}}\right) \quad (10)$$

而对于那些没有被遮掩的词，其预测概率保持不变。

$$y_i^{(t)} = y_i^{(t-1)} \quad (11)$$

$$p_i^{(t)} = p_i^{(t-1)} \quad (12)$$

3 实验

3.1 数据集

我们选择非自回归神经机器翻译任务中常用的 WMT2016 英语-罗马尼亚语[1]语料进行实验，训练集含有 61.3 万对平行句对，验证集 newsdev2016 和测试集 newstest2016 各 2000 对平行句对。我们使用 MOSES[19]对数据进行 tokenize，对英语语料和罗马尼亚语语料进行联合字节对编码[20]（byte pair encoding，BPE），共享约 6 万的词汇表，其余低频词用<UNK>替换。

3.2 实验设置

在实验中，对于长度预测模块，我们使用了一个单层 RNN 来对源端信息进行编码，目标端句子最大长度设为 256，并用一个 Softmax 层对目标端句子长度进行预测。对于目标端句子内容预测部分，我们使用了一个 Transformer 编码器，含有 6 个编码器层，每一层多头注意力机制均使用了 8 个头，词嵌入（Word Embedding）维度为 1024。

训练时，使用 Adam 优化器[21]，初始学习率设为 0.0005，每批次（Batch Size）为 5200 个词，对于所有的隐藏层，都有 0.1 的随机失活率（Dropout），标签平滑（Label Smoothing）参数$\epsilon = 0.1$。和 Lample 等人[10]的实验设置类似，我们随机取样目标端句子中 15%的词，对于这些词，80%的词用[mask]替换，10%的词随机用词表中的词替换，10%的词保持不变。

解码时，我们将迭代优化的次数设为 10。

3.3 翻译系统的评测指标

我们以 BLEU 分数来作为我们模型性能的评测指标，使用 WMT 数据集评测常用的 SacreBLEU[22]工具[2]。

3.4 实验结果

我们的基准系统选取了近年来非自回归神经机器翻译领域比较有影响力的几个工作。分为两类，第一类是解码阶段不需要进行迭代优化的，解码的时间复杂度为$O(1)$；另一类是解码阶段需要进行迭代优化的，解码的时间复杂度为$O(T)$，其中，T 为迭代优化的轮数。这些方法都是基于编码器-解码器结构，具有更多的参数、更复杂的模型结构，会消耗更多的训练时间。

表 1 展示了各系统在 WMT2016 英语-罗马尼亚语任务的测试集 newstest2016 上的性能，可以看出我们的方法尽管结构很简单，但是取得了最好的实验性能（英语→罗马尼亚语 30.2 BLEU，罗马尼亚语→英语 31.2 BLEU）。和解码阶段不需要迭代优化的方法相比，虽然我们的方法在解码时具有更高的时间复杂度，但是我们的模型参数更少，翻译性能更好。而和解码阶段需要迭代优化的方法相比，虽然解码复

[1] http://www.statmt.org/wmt16/translation-task.html

[2] SacreBLEU 的设置: BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.2.17

表 1 各系统在 WMT2016 英语-罗马尼亚语任务的测试集 newstest2016 上的性能

Tab.1 Performance of systems on the test set of WMT2016 English-Romanian task(newstest2016)

系统	BLEU		解码复杂度
	英语→罗马尼亚语	罗马尼亚语→英语	
Fertility-based (Gu 等人, 2018)	27.3	29.1	$O(1)$
CTC (Libovicky 和 Helcl, 2018)	19.5	24.7	
Imitation learning (Wei 等人, 2019)	28.6	28.9	
Reinforcement learning (Shao 等人, 2019)	27.1	27.9	
Generative flow (Ma 等人, 2019)	29.3	30.2	
Extra refinement model (Lee 等人, 2018)	29.3	30.2	$O(T)$
我们的方法	30.2	31.2	

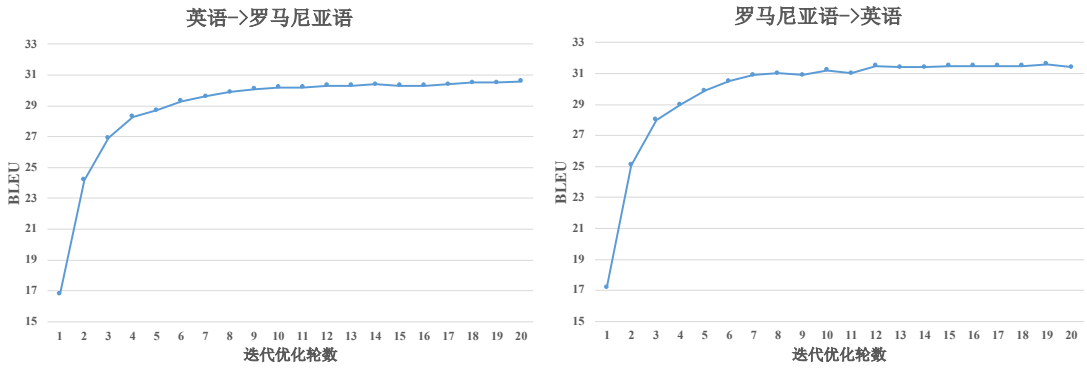

图 3 不同迭代优化轮数对解码结果的影响

Fig.3 Effect of different iteration optimization rounds on the decoding result

杂度相同，但是我们的方法翻译性能有了大幅的提升，而且完全是端到端的训练方式。

4 实验分析

4.1 不同迭代优化轮数的影响

根据 3.6 节介绍的遮掩-预测解码机制，我们可以知道，迭代优化的轮数 N 可能会对解码结果产生一定的影响。而在我们的实验中，迭代优化的轮数 N 已经预先设定，因此，本节我们将分析不同迭代优化轮数对解码结果的影响。

我们从图 3 可以看出，在迭代优化轮数 N 小于 12 时，迭代的轮数越多，模型

表 2 不同的参数初始化方式下模型的性能以及参数量

Tab.2 Performance and parameter quantity of the model with different parameter initialization methods

	BLEU		参数量
	英语→罗马尼亚语	罗马尼亚语→英语	
我们的方法（参数随机初始化）	30.2	31.2	约 100M
我们的方法（使用预训练的 MLM 参数初始化）	33.8	34.5	约 100M
Transformer（使用预训练的 MLM 参数初始化编码器、解码器）	-	35.3	约 200M

解码的效果越好，而在 N 大于 12 时，模型解码的效果趋于平稳，不会因为迭代的次数增多而产生较大的影响。

4.2 模型不同初始化参数的影响

由于我们的方法使用基于掩码机制的 Transformer 编码器，和 MLM 类似，于是我们用预训练的英语-罗马尼亚语 MLM[3] 参数初始化我们的 Transformer 编码器，分析不同初始化参数对模型性能的影响。

由表 2 可知，将我们的 Transformer 编码器用预训练的 MLM 参数初始化之后，英语→罗马尼亚语和罗马尼亚语→英语两个语向上的 BLEU 都取得了很大的提升，同时与 Lample 等人使用预训练的 MLM 参数初始化 Transformer 编码器、解码器得到的自回归神经机器翻译模型性能相当，同时，我们的模型参数约为其的一半，模型更轻量。

4.3 给定目标端长度的表现

在我们的方法中，使用一个单层 RNN 作为长度预测模块。在解码时首先通过长度预测模块预测出目标语言句子的长度，再进行目标语言句子内容的预测。在这里，我们改变解码的方式，即给定目标语言句子的真实长度，然后再用我们的模型预测目标语言句子的内容，分析 RNN 这个长度预测模块的效果。

表 3 给出了相关实验性能的数据，我们在解码阶段给定目标语言句子的真实长度，即不再使用长度预测模块后，我们的解码效果并没有很大的提升（英语→罗马尼亚语方向提升 0.2 BLEU，罗马尼亚语→英语方向提升 0.3），说明我们的长度预测模块在整个模型中起到了不错的效果。

4.4 不同源端信息的影响

如图 2 所示，在我们的 Transformer 编码器中，在对源语言端进行注意力计算的时候，将目标端的信息全部进行了遮掩，因此源语言端每一层的隐状态都只跟源语言端的信息相关，因此我们想到直接用 Transformer 编码器对源语言句子进行独立编码，然后将编码器最终输出的隐状态作为源端信息，参与每一层目标端注意力的计算。

从表 4 可以看出，独立编码源端注意力信息，参与每一层目标端注意力计算的效果不及我们将每一层的源端注意力信息用来参与该层目标端注意力计算。因为每一层的源端隐状态都关注到了不同的信息，参与该层目标端注意力计算能让目标端注意到该层的信息；而独立编码源端注意力信息，将最终输出的隐状态作为源端信息

[3] https://dl.fbaipublicfiles.com/XLM/mlm_enro_1024.pth

参与每一层目标端注意力的计算，会使目标端失去这部分信息，导致效果不佳。

表 3 模型在给定目标端长度情况下的解码效果

Tab.3 Decoding performance of the model with the given target-side length

	BLEU	
	英语→罗马尼亚语	罗马尼亚语→英语
我们的方法	30.2	31.2
我们的方法 +给定目标端长度	30.4	31.5

表 4 独立编码源端注意力信息，参与每一层目标端注意力计算的效果对比

Tab.4 Performance comparison with the method independently encoding the source-side attention information and then participating in the target-side attention calculation of each layer

	BLEU	
	英语→罗马尼亚语	罗马尼亚语→英语
我们的方法	30.2	31.2
独立编码源端注意力信息，参与每一层目标端注意力计算	29.6	30.5

5 结束语

本文针对传统自回归机器翻译结构复杂、参数过多的问题，提出了基于掩码的非自回归神经机器翻译。本文采用类似 MLM 中的掩码机制，对 Transformer 编码器部分修改并使其同时实现编码器与解码器功能。最终实验结果表明我们的基于掩码的非自回归神经机器翻译模型相比于其他非自回归翻译模型，取得了更好的翻译性能；相比传统自回归机器翻译模型，结构更简单、参数更少、训练更快，并且使用跨语言预训练语言模型初始化之后，我们取得了和自回归神经机器翻译模型相当的结果

在未来的工作中，我们将对非自回归机器翻译方法进行进一步探索，同时对我们方法在其他语言对和其他生成任务中的作用进行探索。

参考文献：

[1] 刘洋. 神经机器翻译前沿进展[J]. 计算机研究与发展, 2017(6).

[2] 李亚超, 熊德意, 张民. 神经机器翻译综述[J]. 计算机学报, 2018, 41(12):100-121.

[3] 刘群. 统计机器翻译综述[J]. 中文信息学报, 2003(04):1-12.

[4] KALCHBRENNER N, BLUNSOM P. Recurrent continuous translation models[C]// Empirical Methods in Natural Language Processing. Seattle: EMNLP, 2013: 1700-1709.

[5] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks[C]// Advances in Neural Information Processing Systems. Montreal: NIPS, 2014: 3104-3112.

[6] BAHDANAU D, CHO K, BENGIO Y. Neural machine translation by jointly learning to align and translate[C]// International Conference on Learning Representations. San Diego: ICLR, 2015.

[7] LUONG M T, PHAM H, MANNING C D. Effective approaches to attention-based neural machine translation[C]//Empirical Methods in Natural Language Processing. Lisbon: EMNLP, 2015: 1412–1421.

[8] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems. Long Beach CA: NIPS, 2017: 5998-6008.

[9] DEVLIN J, CHANG M W, LEE K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[C]//North American Chapter of the Association for

Computational Linguistics: Human Language Technologies. Minneapolis: NAACL, 2019: 4171–4186

[10] LAMPLE G, CONNEAU A. Cross-lingual language model pretraining[J]. arXiv preprint arXiv:1901.07291, 2019.

[11] BAPNA A, CHEN M X, FIRAT O, et al. Training deeper neural machine translation models with transparent attention[C]//Empirical Methods in Natural Language Processing. Brussels: EMNLP, 2018: 3028–3033.

[12] HE T, TAN X, XIA Y, et al. Layer-wise coordination between encoder and decoder for neural machine translation[C]//Advances in Neural Information Processing Systems. Montreal: NIPS, 2018: 7944-7954.

[13] FONOLLOSA J A R, CASAS N, COSTA-JUSSÀ M R. Joint Source-Target Self Attention with Locality Constraints[J]. arXiv preprint arXiv:1905.06596, 2019.

[14] GU J , BRADBURY J , XIONG C , et al. Non-Autoregressive Neural Machine Translation[C]//International Conference on Learning Representations. Vancouver: ICLR, 2018.

[15] LEE J, MANSIMOV E, CHO K. Deterministic non-autoregressive neural sequence modeling by iterative refinement[J]. arXiv preprint arXiv:1802.06901, 2018.

[16] GHAZVININEJAD M, LEVY O, LIU Y, et al. Mask-predict: Parallel decoding of conditional masked language models[C]// Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing. Hong Kong: EMNLP-IJCNLP, 2019: 6112–6121.

[17] WEI B, WANG M, ZHOU H, et al. Imitation learning for non-autoregressive neural machine translation[C]// Annual Meeting of the Association for Computational Linguistics. Florence: ACL, 2019: 1304–1312.

[18] SHAO C, FENG Y, ZHANG J, et al. Retrieving sequential information for non-autoregressive neural machine translation[C]// Annual Meeting of the Association for Computational Linguistics. Florence: ACL, 2019: 3013–3024.

[19] KOEHN P, HOANG H, BIRCH A, CALLISON-BURCH C, FEDERICO M, BERTOLDI N, COWAN B, SHEN W, MORAN C, ZENS R, et al. Moses: Open source toolkit for statistical machine translation[C]//Annual meeting of the ACL on interactive poster and demonstration sessions. Prague: ACL, 2007: 177-180.

[20] SENNRICH R, HADDOW B, BIRCH A. Neural machine translation of rare words with subword units[C]// Annual Meeting of the Association for Computational Linguistics. Berlin: ACL, 2016: 1715–1725.

[21] KINGMA D P, BA J. Adam: A method for stochastic optimization[C]// International Conference on Learning Representations. San Diego: ICLR, 2015.

[22] POST M. A call for clarity in reporting BLEU scores[C]//Conference on Machine Translation: Research Papers. Brussels: WMT, 2018: 186–191.

Masking Mechanism for Non-Autoregressive Neural Machine Translation

Abstract: At present, the neural machine translation model based on self-attention mechanism has made great progress. However, the neural machine translation based on autoregressive algorithm can not perform parallel computation in the decoding process, so it takes too much time. We propose

a non-autoregressive neural machine translation model, which can realize parallel computing. Only one encoder module of transformer is used for training, which simplifies the traditional encoder-decoder structure. At the same time, in the training process, we introduce a mask mechanism to reduce the gap between non-autoregressive neural machine translation and autoregressive neural machine translation. Compared with other non-autoregressive translation models, we have achieved better results in WMT 2016 English-Romanian translation tasks, and achieved performances comparable to autoregressive translation models when initialized with cross-lingual pretrained language models.
Keywords: neural machine translation, masked, non-autoregressive