用于大规模数据集语音识别的实时流变压器转换器的研制

*谢晨\**1*，于武\**2*，王正浩*1*，刘淑杰*2*，李金玉*1 1

Microsoft语音和语言组2

微软亚洲研究院

# 摘要

最近，基于变压器的端到端模型在包括语音识别在内的许多领域都取得了巨大的成功。然而，与LSTM模型相比，变压器在推理过程中计算量大是阻碍其应用的一个关键问题。在这项工作中，我们探讨了变压器-换能器（T-T）模型在大规模数据集上实现低延迟快速首通译码的潜力。我们结合TransformerXL和分块流处理的思想设计了一个可流化的Transformer-Transformer模型。我们证明了在流媒体场景中，T-T的性能优于混合模型、RNN-T和基于注意的可流变压器编解码器模型。此外，运行时成本和延迟可以通过相对较小的前瞻性进行优化。

*索引项*-变压器，传感器，实时解码，语音识别

# 1.    导言

端到端（E2E）模式的自动语音识别（ASR）已经取得了很大的进展[1，2，3，4，5，6，7，8]。目前，传感器（例如，递归神经网络传感器（RNNT）[9]）和基于注意的编码器-解码器（AED）[1，10，11]是两种最流行的E2E方法。由于注意机制的存在，AED模型取得了很好的性能，但是它们在本质上不是流式的，有一些朝着这个方向的研究，例如单调的分块注意[12]和触发注意[13，14]。相比之下，由于流的性质，传感器模型特别是RNN-T在工业应用中受到了很大的关注，并且在某些情况下也成功地取代了传统的混合模型[7，8，15]。

虽然带有长-短期记忆（LSTM）[16]单元的RNN在E2E工程中得到广泛应用，但带有自我注意的变压器结构[17]最近已成为E2E模型的基本构建块[18、19、20]。除了准确度之外，流识别器和运行时的计算成本是在工业上部署高质量自动语音识别（ASR）系统的两个关键因素。在这项工作中，我们致力于开发具有高精度和低计算成本的流变压器换能器（T-T）[21]及其变体。变压器变换器的计算量随着输入序列长度的增加而显著增加，这阻碍了T-T的实际应用。最近提出了一致性变换器（C-T）[22]来进一步改进T-T，但由于其编码器对全序列的关注，它不能简化。

现有的方法可能部分地解决了这些问题，但也有自己的缺点。1） 限时法[21，23，24，25]

|  |
| --- |
| \*同等贡献 |

在Transformer中简单地屏蔽左、右上下文以控制时间开销。由于接收场随变压器层数线性增长，该策略引入了较大的延迟。2） 分块方法[26，14]将输入分割成小块，并对每个块进行语音识别。但是，由于忽略了不同语块之间的关系，准确率会下降。3） 基于内存的方法[27，28]使用上下文向量来编码历史信息，同时通过与分块方法相结合来降低运行时成本。然而，该方法打破了变压器在训练中的并行性，需要较长的训练时间。

在本文中，我们的目标是开发可实时操作的流式转换器和一致性转换器模型。我们希望在训练成本、运行时成本和准确性之间达到平衡。我们将Transformer-XL和分块处理相结合来处理流式场景，训练中分块之间没有重叠，保证了训练效率。我们可以在32个V100 gpu上以混合精度在2天内完成6.5万小时的匿名训练数据。在流式语音识别评估中，T-T和C-T的性能优于混合模型、RNN-T和基于流式变换器注意的编解码模型，其相对字错误率（WERR）超过10%。在运行时开销方面，我们提出的方法使用有限的历史，同时保持相同的性能（1%的WERR退化）。如果允许一个小的前瞻，例如360ms，T-T在CPU上达到0.25的实时因子率，满足实际应用的行业要求。

# 2.    模型结构

## 2.1.    传感器结构

在本文中，我们研究了传感器模型[29]在实时和流式语音识别中的应用。一个传感器有三个组成部分，一个声学编码器网络（编码器），一个标签预测器网络（预测器），和一个联合网络。将声学特征序列xis输入编码器，得到编码器输出f*t型*1 *t型*相应地，先前的标签序列y被发送到预测器以计算预测器输出克*u型*−1. 编码器和预测器的输出由联合网络相加。在发送到softmax函数之前，应用非线性函数（如relu函数）来计算句子词汇量的概率分布。传感器的计算公式如下：，

                                        f级=*t型*编码器

                                      克=*u型*−1预测器

                                    h−1=relu（f+−1）*t、 u型t型*克*u型*1

软最大值（Wo\*h）（1）*t、 u型*−1

在实际应用中，我们可以采用不同的编码器和预测器结构。例如，在[29]中，LSTM用于编码器和预测器，并且被广泛地称为RNN-T模型结构。考虑到速度和内存开销，本文采用变压器作为编码器，LSTM作为预测器。

## 2.2.    变压器和合规器

在过去的几年中，变压器模型[17]已经证明在一系列任务[30，31]中比LSTM有显著的性能改进。最近，提出了使用变压器的传感器，并报告其性能优于基于LSTM的传感器模型[31、21、32]。在文献中，为了区分采用RNN作为编码器和解码器的RNN-T，基于变压器的传感器通常被称为变压器传感器（T-T）。

变压器模型采用注意机制捕获序列信息。利用点积相似函数计算输入序列上的注意分布，其形式如下：，



                               =Softmax软件

                        z=XαX*t型tτW型五τ*

*τ*

                                    =Xαv（2）*tττ*

*τ*

式中=√*β*[[1]](" \l "_ftn1" \o ")是一个比例因子。输入向量xis发送到*t型*

*d级*

在注意模块中，分别使用三种不同的矩阵和输出作为查询q、键和值V。在变压器模型中，采用多头部注意（MHA）进一步提高序列模型的容量，对输入序列施加多个并行的自注意，然后将每个注意模块的输出串联起来。等式2中softmax函数的输入序列的范围可以通过应用掩码来控制。如果我们只想使用当前帧和前一帧来计算z，那么注意力权重可以被屏蔽为0。因此，mask的使用为确定输入序列的计算范围提供了一种灵活的方法。本文中使用的掩模设计的细节见第3.2节。在每个变压器层中，它还包含两个完全连接的前馈网络（FFN）、一个非线性激活、层归一化和剩余连接。基于变压器的音频编码器通常堆叠多个变压器层，例如18层。*t型t型t型t型*1 *t型αt，ττ>t*

在变压器模型中，通常采用位置嵌入法对输入序列的排序信息进行显式建模。相对位置嵌入比绝对位置嵌入具有更好的性能[33，30，31]。动机是在计算注意权重时要考虑两帧之间的偏移量，而偏移量是通过相对位置嵌入来建模的。为了提高效率，我们使用了一种简单而有效的相对位置嵌入，其公式如下

z轴=*t型*软最大值*τ*（3） 其中pis是从查找表中获得的相对位置。与文献[30]中使用的相对位置嵌入方法相比，该方法具有更高的效率和内存友好性。*t、 τ*

Transformer模型捕获全局上下文，但局部信息的建模不太好。最近的一些工作[22]表明，CNN和Transformer的结合提高了SR性能。其中，卷积增强变换器（又称Conformer）[22]是一种典型的变换器，它在每个变换器块中插入一种特殊的基于CNN的结构，在Librispeech上实现了最先进的性能。在我们的实现中，我们采用了[22]的原始结构，但是将深度方向的CNN更改为因果深度方向的CNN以避免额外的延迟。

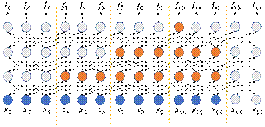
# 3.          实时流变压器-传感器

## 3.1.    流媒体场景中的挑战

在推理过程中，如果使用完整的历史来计算每一帧，计算成本是一个潜在的问题。zin方程2的计算量将随着时间的增加而线性增加，因为它需要计算从第一帧到当前帧的注意权重，这对于长话语来说是负担不起的。此外，在模型精度和延迟之间有一个折衷，如果我们允许多个前瞻帧来计算当前帧，那么性能就可以得到提高。延迟控制技术在混合系统中得到了广泛的应用[34]，最近也被引入到传感器模型中[32，21]。为了使计算变得可行并获得更好的性能，可以对变压器传感器使用截断的历史和有限的未来信息。在[32，21]中，作者设置了注意遮罩，允许在历史和未来的每个转换层中有特定数量的上下文框架。这样可以有效地减少每一层的计算量。但是，对于未来上下文的固定数量来说，存在一个潜在的缺点。随着变换层的增加，未来上下文的数量呈线性增加。如果转换器有18层，并且未来的上下文是每层5帧，那么将引入90帧的延迟。*t型t型*

## 3.2.    流媒体模型训练中的注意遮罩设计

我们设计了一个简单而有效的遮罩策略来截短历史并允许有限的未来信息。Transformer采用注意机制进行序列建模，注意掩码可应用于注意权重矩阵{αt，τ}，以确定计算所涉及的输入序列的范围。当由注意掩码设置为0时，则时间x的输入将不用于计算输出zat time。使用以下规则创建掩码矩阵。首先将输入的声学特征序列分割成具有特定块大小的块，然后根据以下规则构造矩阵：1）同一块中的帧可以相互看到。例如，当计算帧x的编码器输出时，在注意力计算中考虑属于同一块的所有帧，包括xxx。可以看到两个未来帧，而具有零帧展望。每个帧的平均前瞻帧数是块大小的一半。2） 如果两帧处于不同的块中，即使偏移量为1，左帧在注意力计算中也看不到右帧。通过这种方式，块边界将严格限制接收场，避免线性接收场随着模型的加深而增大。3） 如果两个帧位于不同的块中，如果它们的距离小于历史窗口的大小，则右帧可以看到左帧。这种方法使得左接收场随着模型的加深而线性增长（历史窗口大小）。图1b给出了流掩码矩阵的示例。此注意遮罩在所有变压器层上共享。*αt，τττt型t型*1010*,*11*,*12*十*10 *十*12



（a） 位置的接收域。左接收场随变压器层数的增加而增加，而右接收场不随变压器层数的增加而增加。*十*10



（b） 注意每层的遮罩矩阵。如果（i，j）为1，则第th个输入将用于第th帧中的计算。*米米日本我*

图1：我们的遮罩策略的一个例子。在本例中，Chunk size和history window size都是3。

图1a给出了位置的接收字段。我们可以发现，掩蔽策略的优点是允许左侧上下文线性增加，而禁止右侧接收场增长，因此模型可以在限制未来展望的同时使用长历史信息。在该示例中，左接收场每层增长3帧，而右上下文限制为所有层。此外，掩蔽策略非常灵活，它可以模拟我们将在实践中使用的大多数可能的场景。当chunk size为1，history window为无穷大时，它模拟了天真的零前瞻场景。*十*10*十*12

## 3.3.    推理优化

我们使用以下工程优化进行推断。

缓存：给定时间x的输入声帧，为了计算变压器编码器输出f，在每个变压器层中，除了线性矩阵乘法和非线性函数外，我们还需要计算输入序列上的注意权重，然后根据等式2对加权值向量求和。为了避免重复计算，缓存了一些中间变量。具体地说，在等式2中，每个层中的键k=wkx和值v=wvx被缓存一次。因此，我们只需要计算查询q=Wqx，key和value v，然后使用缓存的key和value对输入序列应用softmax函数，用于每个层中的声学帧。另外，由于我们在每个帧中为transformer使用了截断的历史记录，因此缓存的内存消耗不会随着时间的增加而增加。*t型t型t型ττττt型t型t型t型t型t型*

分块计算：如果允许几个帧的延迟，如图1所示，我们可以将输入帧[xxx]分组为一个小的miniback，并馈送到transformer编码器，同时计算ffand fsh。每一层中的密钥和值都将被缓存一次，用于未来帧的计算。这样，可以应用有效的矩阵运算，而不是多次应用矩阵向量运算。值得一提的是，对于具有零前瞻性的transformer编码器，我们可以手动引入几个帧的延迟，并且通过应用这种分块计算可以实现显著的加速。10*,*11*,*1210*,*11 12

# 4.    实验

## 4.1.    实验装置

我们使用了6.5万（K）小时的微软转录数据作为培训数据。测试集涵盖Cortana、远场语音和呼叫中心等13种不同的应用场景，共180万字。报告所有测试场景中的平均字错误率（WER）。所有的训练和测试数据都是匿名数据，删除了个人身份信息。在训练数据转录上训练的4000个句子片段[35]被用作词汇。我们对输入帧使用8的上下文窗口，形成一个640-dim的特征作为传感器编码器的输入，并将帧偏移设置为30ms，从训练数据中丢弃超过30秒的话语。

对于RNN-T，编码器包含6个LSTM层，预测器包含一个嵌入层和2个LSTM层。嵌入层和LSTM层的维度设置为1024。在T-T方面，采用18层720个隐节点和1024个前馈节点作为编码器；与RNN-T相同，采用2层720个隐节点的LSTM作为预测器。C-T选择640作为隐层尺寸，得到与RNN-T和T-T相似的模型尺寸，C-T的核尺寸为3。TT采用固定相对位置编码。所有的模型都是从零开始训练的，并且具有混合精度，以便进行有效的训练。各种传感器型号的模型参数在80M左右。

运行时速度是在一台包含16核的CPU机器上评估的，该机器采用Intel Xeon CPU e5-2620、2.10 GHz和64GB内存。我们从测试集中随机抽取500个话语来测量运行时因素。这些话语的平均长度为12.7秒。浮子精度用于评估LSTM和变压器-传感器模型，在下面的实验中没有明确的说明。对于换能器解码器的波束搜索，所有实验的best设置为5。为了对传感器模型进行有效的解码，采用C++实现了一种基于波束搜索的高效传感器解码器。解码算法的详细信息见[29]。本文使用的传感器模型，包括基于变压器的传感器LSTM，用Pytorch进行训练，然后用Libtorch进行实时（JIT）编译导出。这些JIT导出模型可以方便地、高效地在C++中实现的解码器中进行评估。利用实时因子（RTF）来评价效率。*不*

## 4.2.    评价结果

### 4.2.1.    使用零前瞻设置的评估

由于许多应用程序要求系统对用户的查询做出快速响应，因此零前瞻模型对于实际系统非常重要。因此，第一个实验比较了RNN-T和T-T在1.8M测试集上的性能。所有的模型在未来都看不到任何帧，而是逐帧解码。表1给出了精确度和运行时成本。我们可以观察到，T-T和C-T在准确性上显著优于RNN-T，C-T略优于T-T，这与以往文献一致。然而，在全语境参与下，T-T和C-T的RTF要比1高得多。当我们使用第三节中提出的方法截断左历史时，我们发现模型保持了几乎相同的性能，同时显著降低了RTF。它也是

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | #历史 | WER公司 | RTF（螺纹） | |
|  | #框架 | (%) | 1 | 4 |
| RNN-T公司 | +∞ | 9.86 | 1.56 | 0.46 |
| T-T型 | +∞ | 8.79 | 3.44 | 2.57 |
| T-T型 | 60 | 8.88 | 2.38 | 1.75 |
| C-T公司 | +∞ | 8.78 | 4.02 | 2.56 |
| C-T公司 | 60 | 8.80 | 2.41 | 1.83 |

表1：零前瞻设置的模型比较。

值得注意的是，截断的历史显著减少了内存消耗。完整上下文需要缓存每帧中的所有键和值，并且内存随着音频帧的增加而线性增长，而截断的历史保持键和值向量的固定长度，例如在本实验中为60。结果表明，截断的历史可以在不影响WER性能的前提下有效地降低运行时开销和内存。

然而，即使模型在零前瞻场景中使用有限的上下文，T-T的RTF仍然高于1。我们发现T-T的瓶颈是编码器的运行时间开销，占整个推理阶段的90%左右。逐帧计算非常耗时，因为它没有充分利用变压器模型的并行计算。基于此，我们引入了延迟和运行时成本之间的折衷方法，将多个帧分组形成一个批进行计算。对于RNN-T，我们还可以形成批处理并将其提供给基于LSTM的音频编码器。由于LSTM中的循环连接，它只能部分地并行计算，而且加速比预计要比Transformer慢。表2报告了不同批量大小的RTF。通过较大的批量大小，可以实现更快的解码速度。如果我们为每次计算编码2帧（60毫秒延迟），则RTF小于1。当批量为15时，T-T的RTF可进一步降低到0.2。但是，15个批大小引入了450ms（15\*30ms）的延迟。此外，表2表明变压器比RNN-T更适合批量操作，适当的批量大小使T-T获得与RNN-T相似的RTF。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | #历史长度 | WER公司  (%) |  | RTF（#批量大小） | | |  |
| 1 | 2 | 5 | 10 | 15 |
| RNN-T公司 | +∞ | 9.86 | 0.46 | 0.31 | 0.26 | 0.21 | 0.20 |
| T-T型 | 60 | 8.88 | 1.75 | 0.69 | 0.38 | 0.26 | 0.19 |
| C-T公司 | 60 | 8.80 | 1.83 | 0.95 | 0.48 | 0.36 | 0.25 |

表2:RNN-T、T-T和C-T在零前视训练下累积不同帧数进行批量计算的WER和RTF结果。4个线程用于评估。

### 4.2.2.    使用小型前瞻设置进行评估

根据零前瞻实验，T-T必须用延迟来换取更少的计算成本。对于T-T来说，绝对零前瞻是不可能的，它必须对帧进行批编码。因此，一个想法是用一个小的向前看来做SR，这使得以自然的方式进行分块解码成为可能。T-T和C-T向前看24帧，导致平均延迟为24×30/2=360ms，而RNN-T不采用基于块的解码，其延迟固定为12帧（360ms）。我们还复制了[20]中的数字，以显示测试集上的性能混合系统、可流化Transformer Seq2Seq和离线Transformer Seq2Seq模型，其中混合模型是高度优化的上下文层轨迹LSTM（cltLSTM）[36]，可流化Transformer Seq2Seq基于分块触发注意方法[13，14] ，可流化RNN S2S基于摩卡[12]。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | #历史框架 | #展望（毫秒） | WER公司  (%) | RTF（螺纹） | |
| 1 | 4 |
| 混合的 | +∞ | 480 | 9.34 | - | - |
| RNN S2S公司 | +∞ | 720 | 9.61 | - | - |
| 变速箱。S2S系统 | +∞ | [480, 960] | 9.16 | - | - |
| 变速箱。S2S系统 | +∞ | +∞ | 7.82 | - | - |
| RNN-T公司 | +∞ | 360 | 9.11 | 1.52 | 0.43 |
| T-T型 | 60 | [0,720] | 8.28 | 0.40 | 0.16 |
| C-T公司 | 60 | [0,720] | 8.19 | 0.45 | 0.22 |
| T-T型 | +∞ | +∞ | 7.78 | 0.39 | 0.15 |
| C-T公司 | +∞ | +∞ | 7.69 | 0.36 | 0.15 |

表3:WER和RTF对不同流模型的前瞻性比较。使用混合和S2S模型的第一个块结果来自[20]。

表3给出了不同模型的性能，并给出了一个小的展望。结果表明，由于RNN-T和T-T的性能分别优于可流化RNN S2S和可流化变压器S2S，因此在流化场景中，传感器模型比S2S更强大。T-T和C-T是该场景的更好选择，因为它显示了很强的准确性，同时实现了可接受的运行时成本。一个出乎意料的发现是，T-T在使用整个话语时具有很小的前瞻性，这表明我们简单的T-T策略可以避免与离线模型相比的巨大性能下降。

### 4.2.3.    8位优化

最后的实验比较了INT8量化的效果。INT8量化在保证性能的前提下，有效地降低了内存消耗，加快了推理速度。表4显示了使用带有1个线程的INT8时的WER和speed结果。INT8不会给RNN-t 3.6倍的加速而不影响WER性能。相比之下，INT8在T-T和C-T上引入了轻微的功率退化，产生了大约2倍的加速比。一种可能的解释是，变压器层中的softmax仍然以浮点精度运行，而softmax在CPU上的计算开销很大。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 精密度 | 功率（%） | RTF公司 |
| RNN-T公司 | 浮动32 | 9.11 | 1.56 |
|  | 国际8 | 9.13 | 0.43 |
| T-T型 | 浮动32 | 8.28 | 0.40 |
|  | 国际8 | 8.50 | 0.22 |
| C-T公司 | 浮动32 | 8.19 | 0.45 |
|  | 国际8 | 8.40 | 0.26 |

表4：具有前瞻性的传感器模型的INT8量化的WER和速度结果。

# 5.    结论

我们开发了用于实时语音识别的流式T-T和C-T语音识别模型，希望强大的变压器编码器和流式自然转换器体系结构能够取长补短。我们将Transformer-XL的思想与分块流处理相结合，以避免延迟随着Transformer层的数量线性增长。实验结果表明，在流媒体场景下，T-T和C-T模型的预测精度优于混合模型、RNN-T模型和可流变压器AED模型。在较小的延迟下，T-T和C-T可以获得与RNN-T相当或更好的RTF。

# 6.    参考文献

[1] William Chan、Navdeep Jaitly、Quoc Le和Oriol Vinyals，“倾听、参与和拼写：用于大词汇量会话语音识别的神经网络”，Proc。ICASSP，2016年，第4960-4964页。

[2] R.Prabhavalkar、K.Rao、T.N.Sainath、B.Li、L.Johnson和N.Jaitly，“语音识别序列到序列模型的比较”，过程。Interspeech，2017年，第939-943页。

[3] 陈济同，等，〈探索端到端语音识别的神经转换器〉。ASRU，2017年，第206-213页。

[4] Kanishka Rao，拥有¸im Sak和Rohit Prabhavalkar，“探索RNN传感器流式端到端语音识别的体系结构、数据和单元”，在Proc。ASRU，2017年。

[5] 钟成秋，Sainath，等，，“序列到序列模型的最新语音识别技术”，程序。ICASSP，2018年。

[6] 李杰，叶国庆，大S，赵瑞红，龚妍，“推进声学到词CTC模型”，过程。ICASSP，2018年。

[7] 何彦章，Tara N Sainath，等，“移动设备端到端的流式语音识别”，过程。ICASSP，2019年，第6381-6385页。

[8] 李金玉，赵睿，孟忠，等，“开发RNN-T模型，超越具有定制能力的高性能混合动力模型”，过程。Interspeech，2020年。

[9] A.Graves，“递归神经网络的序列转导”，CoRR，第abs/1211.37112012卷。

[10] Dzmitry Bahdanau、Kyunghyun Cho和Yoshua Bengio，“通过联合学习对齐和翻译的神经机器翻译”，在Proc。ICLR，2015年。

[11] Jan K Chorowski、Dzmitry Bahdanau、Dmitry Serdyuk、Kyunghyun Cho和Yoshua Bengio，“基于注意力的语音识别模型”，2015年，第577-585页。

[12] 钟正秋和柯林拉斐尔，“单调的分块注意力”，在Proc。ICLR，2018年。

[13] Niko Moritz、Takaki Hori和Jonathan Le Roux，“引发了端到端语音识别的关注”，发表于ICASSP 2019年。2019年，第5666-5670页，IEEE。

[14] 王承义、吴宇、刘淑杰、李金玉、陆亮、叶国礼、周明，“利用scout网络降低端到端流式语音识别模型的延迟”。Interspeech，2020年。

[15] Mahaveer Jain，Kjell Schubert，Jay Mahadeokar等人，“改进波束搜索的延迟控制ASR RNNT”，arXiv预印本附件十四：1911.01629, 2019.

[16] S.Hochreiter和J.Schmidhuber，“长-短期记忆”，《神经计算》，第9卷，第8期，第1735-17801997页。

[17] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoret、Llion Jones、Aidan N Gomez、ukasz Kaiser和Illia Polosukhin，《关注是你所需要的一切》，在程序中。NIPS，2017年，第5998-6008页。

[18] 董林浩，徐爽，徐波，“语音转换器：语音识别的无重复序列到序列模型”，过程。ICASSP，2018年，第5884-5888页。

[19] 陈南新，林友基，等，「变压器与RNN在语音应用中的比较研究」，国立台湾大学语言科学研究所。ASRU，2019年。

[20] 李金玉、吴宇、高亚希、王承义、赵睿和刘淑杰，“关于大规模语音识别流行端到端模型的比较”，Proc。Interspeech，2020年。

[21]钱章，韩璐，等，“变压器-换能器：一种具有变压器编码器和rnn-t损耗的可流化语音识别模型”，过程。ICASSP，2020年，第7829-7833页。

[22]Anmol Gulati，James Qin，et al.，“Conformer:用于语音识别的卷积增强变压器”，arXiv预印本附件十四：2005.08100, 2020.

[23]Niko Moritz、Takaki Hori和Jonathan Le Roux，“变压器模型的流式自动语音识别”，ICASSP，2020年。

[24]俞佳慧，韩伟，等，“通用ASR：用全上下文建模统一和改进流式ASR”，arXiv预印本附件十四：2010.06030, 2020.

[25]Anshuman Tripathi，Jaeyoung Kim，Qian Zhang，Han Lu，and Hasim Sak，“Transformer transducer:One model Unified streaming and non streaming speech recognition”，arXiv预印本附件十四：2010.03192, 2020.

[26]田正坤，易江燕，叶白，陶建华，张帅，温正琦，“端到端语音识别的同步变压器”，in Proc。ICASSP公司。IEEE，2020年，第7884-7888页。

[27]吴春阳，王永强，史阳阳，叶清峰，和Frank Zhang，“基于流变压器的声学模型，使用增强记忆的自我注意”，arXiv预印本附件十四：2005.08042, 2020.

[28]Hirofumi Inaguma、Masato Mimura和Tatsuya Kawahara，“增强流式asr的单调多头注意”，arXiv预印本附件十四：2005.09394, 2020.

[29]Alex Graves，“递归神经网络的序列转导”，arXiv预印本附件十四：1211.3711, 2012.

[30]Dai Zihang，Yang Zhilin，Yang Yiming，Jaime Carbonell，Quoc V Le，and Ruslan Salakhutdinov，“Transformer XL:超出固定长度上下文的注意力语言模型”，in Proc。ACL，2019年，第2978-2988页。

[31]Wang Yongqiang，Abdelrahman Mohamed，et al.，“混合语音识别中基于变压器的声学建模”，过程。ICASSP，2020年，第6874-6878页。

[32]叶清峰，Jay Mahadeokar，等，“Transformertransducer:具有自我注意的端到端语音识别”，arXiv预印本附件十四：1910.12977, 2019.

[33]Peter Shaw、Jakob Uszkoret和Ashish Vaswani，“相对位置表示的自我关注”，在Proc。NAACL，2018年。

[34]Yu Zhang，Guoguo Chen，Dong Yu，Kaisheng Yaco，Sanjeev Khudanpur，and James Glass，“用于远程语音识别的高速公路长-短期记忆rnns”，in Proc。ICASSP公司。IEEE，2016年，第5755-5759页。

[35]Taku Kudo，“子词正则化：改进具有多个子词候选的神经网络翻译模型”，ACL，2018，第66-75页。

[36]Jinyu Li，Liang Lu，Changliang Liu，and Yifan Gong，“使用未来上下文框架改进层轨迹lstm”，在Proc。ICASSP公司。IEEE，2019年，第6550-6554页。

[[1]](" \l "_ftnref1" \o ")对于有效的训练，块没有重叠。