On the Comparison of Popular End-to-End Models for Large Scale Speech Recognition

大规模语音识别中常用端到端模型的比较研究

*Jinyu Li*1*，Yu Wu*2*，Yashesh Gaur*1*，Chengyi Wang*2*，Rui Zhao*1*，Shujie Liu*2

1Microsoft语音和语言组

2微软亚洲研究院

# 摘要

Recently, there has been a strong push to transition from hybrid models to end-to-end (E2E) models for automatic speech recognition. Currently, there are three promising E2E methods: recurrent neural network transducer (RNN-T), RNN attention-based encoder-decoder (AED), and Transformer-AED. In this study, we conduct an empirical comparison of RNN-T, RNN-AED, and Transformer-AED models, in both non-streaming and streaming modes. We use 65 thousand hours of Microsoft anonymized training data to train these models. As E2E models are more data hungry, it is better to compare their effectiveness with large amount of training data. To the best of our knowledge, no such comprehensive study has been conducted yet. We show that although AED models are stronger than RNN-T in the non-streaming mode, RNN-T is very competitive in streaming mode if its encoder can be properly initialized. Among all three E2E models, transformer-AED achieved the best accuracy in both streaming and non-streaming mode. We show that both streaming RNN-T and transformer-AED models can obtain better accuracy than a highly-optimized hybrid model.

**Index Terms**: end-to-end, RNN-transducer, attention-based encoder-decoder, transformer

最近，在语音识别中，有一个从混合模式过渡到端到端（E2E）模式的强大推动。目前，有三种很有前途的E2E方法：递归神经网络transducer（RNN-T）、基于attention的RNN编解码器（AED）和Transformer-AED。在这项研究中，我们对RNN-T、RNN-AED和Transformer-AED模型在非流模式和流模式下进行了实验比较。我们使用6.5万小时的微软匿名训练数据来训练这些模型。由于E2E模型更需要数据，因此最好在大量训练数据下对其有效性进行比较。据我们所知，之前从来没有进行过这样全面的研究。我们发现，虽然AED模型在非流式下比RNN-T模型更强大，但如果能正确初始化RNN-T的编码器，RNN-T在流模式下还是非常有竞争力的。在所有三种E2E模型中，transformer-AED在流式和非流式下都取得了最好的准确率。结果还表明，流式RNN-T模型和transformer-AED模型都比高度优化的混合模型具有更好的精度。

**关键词**：端到端，RNN-transducer，attention-based encoder-decoder，transformer

2020.07.29 arvix interspeech 2020.

# 1.     前言

最近，语音界看到了一个重要的趋势，即从基于深度神经网络的混合建模[1]转向端到端（E2E）建模[2，3，4，5，6，7，8，9，10]，用于自动语音识别（ASR）。混合模型需要对单独的组成模型（如声学模型和语言模型）进行独立优化，而E2E-ASR系统则使用单个网络将输入语音序列直接转换为输出令牌（子词，甚至词）序列。

现代广泛使用的E2E序列间转换方法有：（a）连接主义时间分类（CTC）[11，12]，（b）递归神经网络传感器（RNN-T）[13]，和（c）基于attention的编码器-解码器（AED）[14，15，3]。在这三种方法中，CTC是最早的一种，它可以将输入的语音信号映射到目标标签，而不需要任何外部对齐。然而，它也受到帧独立性假设这一条件的影响。RNN-T通过改变目标函数和模型结构来消除帧独立性假设，从而扩展了CTC建模。由于其流式特性，RNN-T在工业应用中受到了广泛关注，并在某些情况下成功地取代了传统的混合模式[9[[1]](#footnote-1)、16[[2]](#footnote-2)、17[[3]](#footnote-3)、18[[4]](#footnote-4)]。

AED是一个通用的模型家族，最初被提出用于机器翻译[19]，但在其他领域（包括ASR[14，15，3]）也取得了成功。默认情况下，这些模型在本质上不是流式的，但是有一些研究朝着这个方向进行改进，比如单调的分块注意[20[[5]](#footnote-5)]和触发注意（triggered attention）[21[[6]](#footnote-6)]。早期的AED模型使用RNN作为其编码器和解码器模块的构建块。在本研究中，我们称之为RNN-AED。最近，具有自我关注（self-attention）的transformer架构[22]也变得流行起来，并被用作编码器和解码器模块的基本构建块[23[[7]](#footnote-7)、24[[8]](#footnote-8)、25[[9]](#footnote-9)]。本文将这种模型称为transformer-AED。

鉴于E2E技术的快速发展前景，比较该领域最流行和最有前途的E2E技术是急需的，从而形成未来的研究方向。本文重点比较了当前最有前途的E2E技术，即RNN-T、RNN-AED和Transformer-AED在非流模式和流模式下的性能。所有模型都使用了6.5万小时的微软匿名训练数据。由于E2E模型需要大量的数据，因此最好在如此大量的训练数据下，对其能力进行比较。据我们所知，目前还没有这样详细的比较研究。在最近的一项工作[162]中，将流式RNN-T模型与非流式RNN-AED进行了比较。在[26[[10]](#footnote-10)]中，流式RNN-AED与流式RNN-T在长格式语音识别中进行了比较。在[259]中，RNN-AED和Transformer-AED在非流模式下进行比较，训练数据为960小时。由于工业应用通常需要流模式下的ASR服务，我们进一步研究了如何在流模式下开发这些E2E模型。虽然[27[[11]](#footnote-11)]中已经表明，在二次解码配置中结合RNN-T和RNN-AED可以超过行业级最先进的混合模型，但本研究表明，单个流式E2E模型（RNN-T或Transformer -AED）也可以超过最先进的混合模型[28[[12]](#footnote-12)，29[[13]](#footnote-13)]。

除了第一次对这些有前途的E2E模型进行详细的比较外，本文的其他贡献是：（1）提出了一种多层上下文建模方案，以使用将来的上下文，并取得了显著的成果；2） 交叉熵（CE）初始化比CTC初始化更有效地提高了RNN-T模型的性能；3）对于流式transformer-AED，基于块的未来上下文方案，比lookahead方法更有效；4） 我们在Librispeech的[30]上发布了与transformer相关的代码，其结果是可复现的，以便于将来的研究。

# 2.     流行的端到端模型

在本节中，我们简要介绍当前流行的E2E模型：RNN-T、RNN-AED和Transformer-AED。这些模型有一个产生高级语音表示的声学编码器（encoder），以及一个自回归产生语言域输出标记的解码器（decoder）。RNN-T和AED声学编码器可以是相同的，但解码器是不同的。在RNN-T中，下一个标签的产生只取决于前一步的标签输出，而AED的解码器的下一个输出还依赖于声学模块。更重要的是，RNN-T以帧同步方式工作，而AED以标签同步方式工作。

## 2.1.    RNN传感器

编码器网络转换声学特征*x*1： T成为高级表示形式henc1:T。译码器，称为预测网络，产生一个高级表示*hpreu*通过消耗以前的非空目标yu-1。这里u表示输出标签索引。联合网络是一种前馈网络，它将编码器网络输出h和预测网络输出h相结合，生成用于计算softmax输出的联合矩阵h。这里表t示时间索引。

编码器和预测网络通常使用带有LSTM[31]单元的RNN来实现。当编码器是单向LSTM-RNN时，如等式（1）所示，RNN-T默认在流模式下工作。

*henct = LSTM*(*xt, henct-1*) (1)

然而，当底层LSTM-RNN编码器是如等式（2）所示的双向模型时，它是非流E2E模型。

*henct =*[ *LSTM*(*xt, henct-1*)，*LSTM*(*xt, henct+1*) ] (2)

当预测网络使用LSTM-RNN实现时公式是

*hencu = LSTM*(*yu-1, hpreu-1*)(3)

利用transformer模型的优势，最近的一项工作是用transformer模型取代编码器中的LSTM-RNN，以构建transformer-transducer [32]和conformer transducer[33]。

## 2.2.    基于attention的编解码器

RNN-T由于其流的特性受到了业界的广泛关注，而基于attention的编码器（AED）模型由于其强大的注意结构而受到学术界的广泛关注。RNN-AED和Transformer-AED分别采用LSTM-RNN和Transformer实现编码器和解码器。

### 2.2.1.    RNN-AED

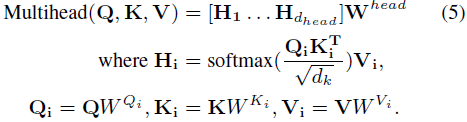
RNN的编码器可以具有与RNN-T相同的结构，如式（1）和式（2）。然而，使用attention增强的decoder操作不同，如下所示：

*hdecu = LSTM*(*cu,* *yu-1, deceu-1*)(4)

这里cu是通过编码器输出的加权组合获得的上下文向量。cu应该包含发出下一个标记所需的声音信息。它是利用注意机制计算出来的[14，34]。

### 2.2.2.    transformerAED

即使RNN可以捕获长期依赖，基于Transformer[22]的模型也可以更有效地捕获，因为注意机制可以直接看到所有上下文。具体而言，编码器由transformer块的堆栈组成，其中每个块具有多头自我注意（multi-head self-attention）层和前馈层。假设transformer块的输入可以线性变换为Q, K,和V。那么，multi-head self-attention的输出是



这里dhead是注意头的数量，dk是每个头的特征向量的维数。该输出送到前馈层。当我们连接不同的层和块时，残差连接[35]和层规范化（LN）[36]是必不可少的。除了编码器块中的两层之外，transformer decoder还具有附加的第三层，该第三层对编码器的输出执行多头注意。这与RNN-AED的注意机制相似。

# 3.     我们的模型

## 3.1.    模型构建块

E2E模型的编码器和解码器，由本节中描述的多个模块（building blocks）的堆叠构建而成。对于使用LSTM-RNN的模型，我们探讨了两种结构。第一个是lstm\_cudnn，它直接调用nvidia cudnn库[37]来实现LSTM。我们通过连接一个cuDNN LSTM层、一个线性投影层来减小模型尺寸，然后连接LN来构建每个块。基于nvidia cudnn实现，使我们能够快速地比较不同的模型。

第二个结构是LSTM \_Custom，它将LN和投影层放在LSTM中，正如[9]中所指出的，它们对于提高RNN-T模型训练非常重要。因此，我们仅通过定制LSTM函数将此结构用于RNN-T。详细描述见[17]。但是，这会导致模型训练速度降低50%。

对于Transformer AED模型，我们移除了位置嵌入部分[38]，并在Transformer块之前，使用类似VGG的卷积模块[39]对语音特征x1:t进行预处理。将LN置于多头注意层（Pre-LN）之前，使得梯度在训练初期表现良好。

## 3.2.    非流媒体模型

通过在编码器中添加双向性，实现了RNN-T中的非流行为。该RNN-T的编码器由多个双向LSTM\_cuDNN块组成，如第3.1节所述。预测网络由LSTM\_cuDNN的多个单向块实现。

与RNN-T类似，本文研究的非流RNN-AED也在编码器中使用了多个双向LSTM\_cuDNN块，在解码器中使用了单向LSTM\_cuDNN块。该解码器与location-aware softmax attention一起工作[15]。RNN-AED不采用联合CTC的多任务训练或联合解码。

参考[25]，Transformer-AED模型采用了多任务训练和CTC/attention联合解码。训练目标函数为

L=−αlog*pctc*(9y | x1:T)−（1−α）log*patt*(y | x1:T)(6)

联合译码中下一个子字log*p*(*yu | x1:t, y1:u*)的对数似然表示为

log*pctc*(*yu | x1:t, y1:u*)+β1 log*patt*(*yu | x1:t, y1:u*)。(7)

在实践中，我们首先使用注意模型来选择前k候选结果，然后用公式7对他们重新排序。

## 3.3.    流媒体模型

流RNN-T模型有一个单向编码器。我们可以直接采用3.1节描述的LSTM\_cuDNN或LSTM\_Custom这类LSTM作为构建块；此外将未来上下文合并到编码器结构中可以显著提高ASR精度，如[17]所示。然而，与文献[17]探讨未来的上下文框架和层轨迹结构不同，在本研究中，我们建议只使用上下文建模。我们这样做是为了节省模型参数。使用下面的简单公式来模拟未来上下文。



因为⊙是元素积，所以式（8）只略微增加了模型参数的个数。它传递一个低层向量*gtl*和它的未来向量一起变成一个新的向量*ζtl*，其中*δ*是未来帧索引。我们使用上下文建模来修改LSTM\_cuDNN或LSTM Custom的块。

•LSTM\_cuDNN\_Context：该块由Nvidia cuDNN LSTM层构造，然后是线性投影层，然后是上下文建模层，最后是LN层。

•LSTM\_Custom\_Context：该块是用带投影的layer normalized LSTM层构造的，然后是context建模层。

在[40]中，将类似的上下文建模概念应用于RNN，作为先行卷积层（Lookahead convolution layer）。然而，它只用于多层RNN的顶层。相反，在本研究中，我们将上下文建模应用于LSTM\_cuDNN或LSTM Custom的每一个块，并探讨其在E2E建模中的有效性。对于RNN-T，我们还研究了用CTC[6]或CE训练[41]初始化编码器。

RNN-AED模型使用LSTM\_cuDNN\_Context块作为编码器。LSTM\_Custom\_Context的实验将是未来研究的一部分。我们在本研究中选择的流机制是单调的分块注意（MoChA）[42]。MoChA由一个单调的注意机制组成[43]，它以从左到右的顺序扫描编码器输出，并在决定触发解码器时选择特定的编码器状态。这个选择概率是通过从一个参数化的伯努利随机变量中抽样来选择的。一旦检测到触发点，MoChA还会使用一个额外的回望窗口，并对其应用常规的softmax关注。注意，我们这里有一个采样操作，这致使标准反向传播无法使用。因此，我们根据上下文向量的期望值进行训练。详情请参阅[42]。

为了在Transformer-AED模型中实现流式场景，我们借用了trigger attention（TA）[21]中的思想，CTC进行帧同步解码，为每一帧选择前k个候选，然后一旦CTC触发一个新的子词，利用注意模型使用等式7对候选进行联合重新排序。由于transformer编码器比LSTM更深，因此lookahead方法可能不是最佳解决方案。我们比较了chunk-based方法和lookahead-based方法。前者将整个输入分为若干固定长度的块，然后逐块输入到模型中，后者与RNN-T和RNN-AED中的方法完全相同。对于基于块的编码器，解码器可以看到块的结尾。对于基于lookahead的编码器，我们为解码器设置了一个固定的窗口大小。

# 4.     实验

在本节中，我们使用6.5万（K）小时的Microsoft转录数据对所有模型进行训练，评估其有效性。测试集涵盖Cortana和远场语音等13种应用场景，共包含180万（M）字。我们报告了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的，个人识别信息被删除。

为了公平比较，所有为本研究建立的E2E模型都有大约87m的参数。输入特征是80维log Mel滤波器组，步长为10毫秒（ms）。其中三个叠在一起形成一个240维的超级帧。特征送入RNN-T和RNN-AED的编码器网络，而transformer-AED直接使用10ms特征。所有E2E型号都使用相同的4K字元（word piece）作为输出目标。

## 4.1.    非流E2E模型

如第3.1节所述，非流RNN-T模型使用双向LSTM，其编码器中带有Nvidia cuDNN库。LSTM memory单元大小为780。向前和向后方向的LSTM输出合并成为1560维，然后线性投影到780维，接着是LN层。此类操作共有6个堆叠块。预测网络有2个堆叠块，每个块包含一个单向cuDNN LSTM，存储单元大小为1280，然后是一个线性投影层，将维数降低到640，然后是LN层。

非流RNN-AED模型使用与非流RNN-T模型完全相同的编码器和解码器结构。与[34]类似，使用了位置感知注意机制。除了编码器和解码器的隐藏状态外，该机制还将来自先前解码器步骤的对齐作为输入。attention维度为512。

Transformer-AED模型在编码器中有18个transformer块，在解码器中有6个transformer块。在编码器中transformer前面，采用四层VGG网络对语音特征进行预处理，总步长为4。注意头数为8个，每个注意头的注意维数为64。在transformer组中，前馈层的尺寸为2048。联合训练和解码（即*α,β*）的组合权重均为0.3。

表1所有非流E2E模型在13个测试集上的平均WER，其中包含1.8 M单词。

|  |  |
| --- | --- |
| 非流式模型 | WER |
| RNN-T（cuDNN） | 9.25 |
| RNN-AED（cuDNN） | 8.05 |
| Transformer-AED | 7.83 |

如表1所示，由于注意力建模的力量，非流式AED模型比非流式RNN-T模型具有明显的优势。Transformer-AED使RNN-AED相对WER降低2.7%。

## 4.2.    流E2E模型超越混合模型

在[28]中，我们报告了称为上下文层轨迹LSTM（cltLSTM）[29]的最佳混合模型的结果。cltLSTM采用三阶段优化过程进行训练。该模型的WER比CE基线有16.2%的相对下降。引入总计24帧的未来上下文，WER进一步有18.7%的相对下降。编码延迟仅为480毫秒（24\*20毫秒=480毫秒；由于跳帧，每帧的步幅为20毫秒[44]）。因此，这个cltLSTM模型（表2）提出了一个非常具有挑战性的流混合模型。该模型具有65m的参数，采用5gram解码图进行解码。

我们在表2中列出了所有流E2E模型的结果。基线RNN-T编码器和解码器在实现时，使用单向cuDNN LSTM。编码器有6个LSTM\_cuDNN块。每个块都有一个单向cuDNN LSTM，1280个存储单元，然后投影到640维，后跟LN。预测和联合网络与非流RNN-T模型相同。该RNN-T模型获得了12.16%的WER。第二个RNN-T模型在每个块中的线性投影层之后插入上下文建模层（等式（8））。上下文建模在每个块上有4帧lookahead，因此编码器有4\*6=24帧lookahead。由于帧移为30ms，编码器的总前向时间为720ms，lookahead的wer提高到了10.65%。与第一个没有lookahead的RNN-T模型相比，WER相对下降了12.4%。我们还遵循了[40]中提出的前瞻卷积（lookahead convolution），只在最顶层的RNN块上使用24帧前瞻。这个模型给出了11.19%的WER，这表明我们提出的上下文建模，它在每个块上平均分配先行帧，比先行卷积（lookahead convolution）[40]要好，后者只将所有先行帧放在顶层。

接下来，我们看一下编码器初始化对RNN-T的影响。如表2所示，RNN-T编码器的CTC初始化没有太大帮助，而CE初始化显著地将WER降低到9.80。与随机初始化模型相比，WER相对下降了8.0%。CTC初始化使编码器输出令牌尖峰和大量空白，而CE初始化使编码器学习时间对齐。考虑到CE初始化时的增益，我们认为RNN-T编码器的功能更像混合模型中的声学模型。注意，CE预训练需要时间对齐，这对于单字单元很难获得，因为它们中的许多没有音素实现。但是，单词的时间对齐仍然是准确的。我们通过简单地将一个词的持续时间平均地分割成它的组成词段，来近似并获得一个词段的对齐。

对于最后一个RNN-T模型，我们将投影层和LN放在LSTM单元（CustomLSTM）中，然后在其后面插入上下文建模层。把投影层放在里面可以让我们使用更多的内存单元，同时保持与cuDNNLSTM设置相似的模型大小。这个LSTM有2048个内存单元，项目层将输出大小减少到640。这个模型最终给出了9.27%的功率，这比我们最好的混合模型稍微好一点。

基于MoChA的流式RNN-AED模型采用与cuDNN-RNN-T相同的编码器架构，结果令人印象深刻。与RNN-T不同的是，它不需要任何初始化，在一对一的比较中仍然能够略胜于RNN-T（9.61%对9.80%）。据我们所知，这是流式RNN-AED第一次超过大规模任务的RNN-T。请注意，我们之前的研究没有观察到RNN-AED与CE初始化的准确性提高[45]。我们将在未来的研究中探讨RNN-AED是否也能从customized LSTM功能中获益。

表2：流式模型在13个包含180万个单词的测试集上的平均WER。

|  |  |  |
| --- | --- | --- |
| Streaming models | WER | Encoder lookahead |
| Hybrid cltLSTM | 9.34 | 480毫秒 |
| RNN-T  cuDNN | 12.16 | 0毫秒 |
| cuDNN+Context | 10.65 | 720毫秒 |
| cuDNN+convolution[40] | 11.19 | 720毫秒 |
| cuDNN+Context+CTC init. | 10.62 | 720毫秒 |
| cuDNN+Context+CE init. | 9.80 | 720毫秒 |
| Custom+Context+CE init. | 9.27 | 720毫秒 |
| RNN-AED  cuDNN+Context | 9.61 | 720毫秒 |
| transformerAED  lookahead method | 10.26 | 720毫秒 |
| Chunk-based method | 9.16 | 720毫秒 |

流transformer-AED模型的结构与非流transformer-AED模型相同。对于lookahead context modeling方法，每个编码器块向前看1帧。考虑到VGG的总步幅为4，语音采样率为10ms，编码器的延迟为1\*18\*4\*10ms=720ms。lookahead方法的解码器引入了额外的240ms延迟。基于块的方法考虑具有固定块的未来上下文。每个帧的延迟在[480ms 960ms]的范围内，导致平均延迟720ms而没有额外的解码器延迟。基于块的方法获得了9.16%的WER，显著优于lookahead方法，这主要是因为lookahead方法的底层transformer 块不能享受正确上下文提供的全部优势。*,*

# 5.     结论

本文首次对三种流行的E2E模型（RNN-T、RNN-AED和Transformer-AED）进行了大规模的比较研究。在流模式和非流模式下对模型进行了比较。所有模特都使用6.5万小时的微软内部匿名数据进行训练。我们观察到，在相同的编码器结构下，无论对于非流式模型还是流式模型，AED都优于RNN-T。在使用customized LSTM作为编码器并进行CE初始化后，RNN-T模型优于RNN-AED。在所有模型中，Transformer-AED在流模式和非流模式下均获得最佳的WER。

在这项研究中，流式RNN-T和Transformer-AED都优于高度优化的混合模型。有几个重要因素促成了这一成功。对于流式RNN-T，本文提出的上下文模型相对于没有任何lookahead的模型降低了12.4%的WER。RNN-T的CE初始化比基线的随机初始化WER相对下降8.0%。这表明，即使在大规模的任务中，预先训练也是有帮助的。在流式transformer-AED上使用未来上下文，我们证明了chunk-based方法比lookahead方法好10.7%。

# 6.     References

[1] G. Hinton, L. Deng, D. Yu et al., “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82–97, 2012.

[2] Y. Miao, M. Gowayyed, and F. Metze, “EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding,” in Proc. ASRU. IEEE, 2015, pp. 167–174.

[3] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in Proc. ICASSP, 2016, pp. 4960–4964.

[4] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, “A comparison of sequence-to-sequence models for speech recognition,” in Proc. Interspeech, 2017, pp. 939–943.

[5] E. Battenberg, J. Chen, R. Child, A. Coates, Y. G. Y. Li, H. Liu, S. Satheesh, A. Sriram, and Z. Zhu, “Exploring neural transducers for end-to-end speech recognition,” in Proc. ASRU. IEEE, 2017, pp. 206–213.

[6] K. Rao, H. Sak, and R. Prabhavalkar, “Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer,” in Proc. ASRU, 2017.

[7] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J.Weiss, K. Rao, K. Gonina et al., “State-of-the-art speech recognition with sequence-to-sequence models,” in Proc. ICASSP, 2018.

[8] J. Li, G. Ye, A. Das, R. Zhao, and Y. Gong, “Advancing acoustic-to-word CTC model,” in Proc. ICASSP, 2018.

[9] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang et al., “Streaming end-to-end speech recognition for mobile devices,” in Proc. ICASSP, 2019, pp. 6381–6385.

[10] J. Li, , R. Zhao, Z. Meng et al., “Developing RNN-T models surpassing high-performance hybrid models with customization capability,” in Proc. Interspeech, 2020.

[11] A. Graves, S. Fern´andez, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in Proceedings of ICML. ACM, 2006, pp. 369–376.

[12] A. Graves and N. Jaitley, “Towards end-to-end speech recognition with recurrent neural networks,” in PMLR, 2014, pp. 1764–1772.

[13] A. Graves, “Sequence transduction with recurrent neural networks,” CoRR, vol. abs/1211.3711, 2012.

[14] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” arXiv preprint arXiv:1409.0473, 2014.

[15] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” in NIPS, 2015, pp. 577–585.

[16] T. Sainath, R. Pang, and et. al., “Two-pass end-to-end speech recognition,” in Proc. Interspeech, 2019.

[17] J. Li, R. Zhao, H. Hu, and Y. Gong, “Improving RNN transducer modeling for end-to-end speech recognition,” in Proc. ASRU, 2019.

[18] M. Jain, K. Schubert, J. Mahadeokar et al., “RNN-T for latency controlled ASR with improved beam search,” arXiv preprint arXiv:1911.01629, 2019.

[19] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” arXiv preprint arXiv:1409.0473, 2014.

[20] C.-C. Chiu and C. Raffel, “Monotonic chunkwise attention,” arXiv preprint arXiv:1712.05382, 2017.

[21] N. Moritz, T. Hori, and J. Le Roux, “Triggered attention for endto- end speech recognition,” in Proc. ICASSP, 2019, pp. 5666– 5670.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in Advances in Neural Information Processing Systems, 2017, pp. 6000–6010.

[23] L. Dong, S. Xu, and B. Xu, “Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition,” in Proc. ICASSP, 2018, pp. 5884–5888.

[24] S. Zhou, L. Dong, S. Xu, and B. Xu, “Syllable-based sequence-to-sequence speech recognition with the transformer in Mandarin Chinese,” in Proc. Interspeech, 2018.

[25] S. Karita, N. Chen, T. Hayashi et al., “A comparative study on transformer vs RNN in speech applications,” in Proc. ASRU, 2019.

[26] C.-C. Chiu, W. Han, Y. Zhang et al., “A comparison of end-to-end models for long-form speech recognition,” in Proc. ASRU, 2019.

[27] T. N. Sainath, Y. He, B. Li et al., “A streaming on-device end-toend model surpassing server-side conventional model quality and latency,” in Proc. ICASSP, 2020, pp. 6059–6063.

[28] J. Li, R. Zhao, E. Sun, J. H.Wong, A. Das, Z.Meng, and Y. Gong, “High-accuracy and low-latency speech recognition with twohead contextual layer trajectory LSTM model,” in Proc. ICASSP, 2020.

[29] J. Li, L. Lu, C. Liu, and Y. Gong, “Improving layer trajectory LSTM with future context frames,” in Proc. ICASSP, 2019, pp. 6550–6554.

[30] C. Wang, Streaming Transformer, 2020.[Online]. Available: https://github.com/cywang97/StreamingTransformer

[31] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[32] Q. Zhang, H. Lu, H. Sak et al., “Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss,” in Proc. ICASSP, 2020.

[33] A. Gulati, J. Qin, C.-C. Chiu, et al., “Conformer: Convolutionaugmented transformer for speech recognition,” arXiv preprint arXiv:2005.08100, 2020.

[34] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, “End-to-end attention-based large vocabulary speech recognition,” in Proc. ICASSP. IEEE, 2016, pp. 4945–4949.

[35] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” arXiv preprint arXiv:1512.03385, 2015.

[36] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” arXiv preprint arXiv:1607.06450, 2016.

[37] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer, “cuDNN: Efficient primitives for deep learning,” arXiv preprint arXiv:1410.0759, 2014.

[38] C.Wang, Y.Wu, Y. Du, J. Li, S. Liu, L. Lu, S. Ren, G. Ye, S. Zhao, and M. Zhou, “Semantic mask for transformer based end-to-end speech recognition,” in Proc. Interspeech, 2020.

[39] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. ICLR, 2015.

[40] C. Wang, D. Yogatama, A. Coates, T. Han, A. Hannun, and B. Xiao, “Lookahead convolution layer for unidirectional recurrent neural networks,” in Proc. ICLR Workshop, 2016.

[41] H. Hu, R. Zhao, J. Li, L. Lu, and Y. Gong, “Exploring pretraining with alignments for RNN transducer based end-to-end speech recognition,” in Proc. ICASSP, 2020.

[42] C.-C. Chiu\* and C. Raffel\*, “Monotonic chunkwise attention,” in International Conference on Learning Representations, 2018.

[43] C. Raffel, D. Eck, P. Liu, R. J. Weiss, and T. Luong, “Online and linear-time attention by enforcing monotonic alignments,” in Thirty-fourth International Conference on Machine Learning, 2017.

[44] Y. Miao, J. Li, Y. Wang, S. Zhang, and Y. Gong, “Simplifying long short-term memory acoustic models for fast training and decoding,” in Proc. ICASSP, 2016.

[45] H. Inaguma, Y. Gaur, L. Lu, J. Li, and Y. Gong, “Minimum latency training strategies for streaming sequence-to-sequence asr,” in Proc. ICASSP, 2020.

1. *Streaming end-to-end speech recognition for mobile devices*. 2019 [↑](#footnote-ref-1)
2. *Two-pass end-to-end speech recognition*. 2019 [↑](#footnote-ref-2)
3. *Improving RNN transducer modeling for end-to-end speech recognition*. 2019 [↑](#footnote-ref-3)
4. *RNN-T for latency controlled ASR with improved beam search*. 2019 [↑](#footnote-ref-4)
5. *Monotonic chunkwise attention*. 2017 [↑](#footnote-ref-5)
6. *Triggered attention for endto- end speech recognition*. 2019 [↑](#footnote-ref-6)
7. *Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition*. 2018 [↑](#footnote-ref-7)
8. *Syllable-based sequence-to-sequence speech recognition with the transformer in Mandarin Chinese*. 2018 [↑](#footnote-ref-8)
9. *A comparative study on transformer vs RNN in speech applications*. 2019 [↑](#footnote-ref-9)
10. *A comparison of end-to-end models for long-form speech recognition*. 2019 [↑](#footnote-ref-10)
11. *A streaming on-device end-toend model surpassing server-side conventional model quality and latency*. 2020 [↑](#footnote-ref-11)
12. *High-accuracy and low-latency speech recognition with twohead contextual layer trajectory LSTM model*. 2020 [↑](#footnote-ref-12)
13. *Improving layer trajectory LSTM with future context frames*. 2019 [↑](#footnote-ref-13)