Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss

Transformer transducer：一个使用transformer和rnn-t损失的可流式语音识别模型

*Qian Zhang, Han Lu, Hasim Sak, Anshuman Tripathi, Erik McDermott, Stephen Koo, Shankar Kumar*

Google Inc., USA

# 摘要

In this paper we present an end-to-end speech recognition model with Transformer encoders that can be used in a streaming speech recognition system. Transformer computation blocks based on selfattention are used to encode both audio and label sequences independently. The activations from both audio and label encoders are combined with a feed-forward layer to compute a probability distribution over the label space for every combination of acoustic frame position and label history. This is similar to the Recurrent Neural Network Transducer (RNN-T) model, which uses RNNs for information encoding instead of Transformer encoders. The model is trained with the RNN-T loss well-suited to streaming decoding. We present results on the LibriSpeech dataset showing that limiting the left context for self-attention in the Transformer layers makes decoding computationally tractable for streaming, with only a slight degradation in accuracy. We also show that the full attention version of our model beats the-state-of-the art accuracy on the LibriSpeech benchmarks. Our results also show that we can bridge the gap between full attention and limited attention versions of our model by attending to a limited number of future frames.

Index Terms— Transformer, RNN-T, sequence-to-sequence, encoder-decoder, end-to-end, speech recognition

本文提出了一种可用于流式语音识别系统的使用Transformer encoder的端到端语音识别模型。使用基于自关注的transformer计算块对音频序列和标签序列进行独立编码。来自音频和标签编码器的激活通过前馈层相结合，以计算声学帧位置和标签历史的每个组合在标签空间上的概率分布。这类似于递归神经网络传感器（RNN-T）模型，该模型使用RNN代替Transformer编码器进行信息编码。该模型采用适合于流解码的RNN-T loss进行训练。我们在LibriSpeech数据集上给出的结果表明，在Transformer层中限制用于自我注意的左上下文使得解码在计算上易于处理，而在精确度上只有轻微的降低。我们还表明，我们模型的完全关注（full attention）版本在LibriSpeech基准上优于最先进的准确性。我们的结果还表明，我们可以通过关注有限数量的未来帧来弥补完全注意和有限注意版本之间的差距。

关键词：transformer，RNN-T，序列到序列，编解码器，端到端，语音识别

2020.02.14 arxiv, icassp 2020

# 1.    导言

在过去的几年中，采用自我注意(self-attention)的模型已经在许多任务上取得了最新的成果，例如机器翻译、语言建模和语言理解[1，2]。特别是，基于大型transformer的语言模型在用于二次扫描和一次浅层融合时，在语音识别任务中带来了收益[3]。正如通常用于序列到序列转换任务[4、5、6、7、8]中一样，基于transformer的模型使用解码器特征关注所有编码器特征，这意味着解码必须以标签同步方式完成，从而对流式语音识别应用提出了挑战。使用这些模型进行流式语音识别的另一个挑战是，自我注意的计算量随输入序列大小的平方增加。为了流式传输在计算上是实用的，处理每一帧所需的时间相对于输入的长度保持恒定是非常理想的。基于transformer的RNN替代方案最近被探索用于ASR[9，10，11，12]。

对于流式语音识别模型，递归神经网络（RNN）已经成为事实上的选择，因为它们可以有效地建模音频特征的时间依赖性[13] 同时保持每个帧的恒定计算需求。可流化的端到端建模架构，如递归神经网络传感器（RNN-T）[14、15、16]、递归神经对齐器（RNA）[17]和神经传感器[18]利用基于编码器-解码器的框架，其中编码器和解码器都是RNN层，分别从音频和标签生成特征。特别地，RNN-T和RNA模型被训练来学习声学编码器特征和标签编码器特征之间的对齐，因此自然地适合于帧同步解码。

|  |
| --- |
|  |

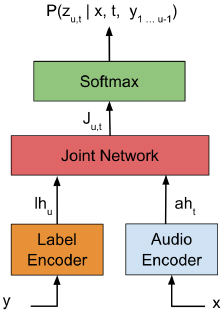


图1。RNN/transformer transducer结构。

为了在设备上运行RNN-T，[16]评估了几种优化技术。此外，[15]对RNN-T进行了广泛的架构和建模单元探索。在本文中，我们探讨了用transformer编码器取代传统RNN-T架构中基于RNN的音频和标签编码器的可能性。为了保持模型的可流性，我们证明了基于transformer的模型可以在固定数量的过去输入帧和以前的标签上进行自我注意训练。这会导致性能下降（与处理所有过去的输入帧和标签相比），但是模型满足处理每个帧的恒定计算需求，因此适合流式传输。考虑到自注意计算的简单结构和可并行性，我们观察到与使用RNN的RNN-T模型相比，在训练时间和训练资源利用率方面有很大的改进。

RNN-T架构（如图1所示）是一种神经网络架构，可以使用RNNT损失端到端地训练该架构，以将输入序列（例如音频特征向量）映射到目标序列（例如音素、图形）。给定长度实值向量的输入序列*T，***x**=（x1，x2，…，xT），RNN-T模型试图预测*U*长度标签**y**=（y1，y2，…，yU）的目标序列。

与典型的基于attention的序列到序列模型不同，它们对输出序列中的每个预测都关注整个输入，RNN-T模型在每个时间步都给出了标签空间上的概率分布，并且输出标签空间包括一个额外的空标签，以指示该时间步长缺少输出-类似于连接主义时间分类（CTC）框架[19]。但与CTC不同的是，这种标签分布也取决于以前的标签历史。

RNN-T模型定义了所有可能路线的条件分布P（z | x），其中

z = [(z1; t1); (z2; t2); :::; (zU; tU)]

是长度为*U‘*的（zi，ti）序列对，这里（zi，ti）表示输出标签zi和编码特征在时间ti上的对齐。标签zi可以是空白标签（空预测）。删除空白标签给出实际的输出标签序列y，长度为*U*。

我们可以在所有可能的对齐上估计P（z | x），以获得给定输入序列y的目标标签序列的概率，



其中Z（y，T）是长度为*T*的有效对齐标签序列集合。

# 2.    transformer transducer

## 2.1.    RNN-T体系结构和损失

在本文中，为了保持一致性，我们给出的所有实验均使用RNN-T损失[14]，其表现类似于我们[2]实验中的单调RNN-T损失。

一个对齐P（z | x）的概率可以分解为



其中Labels（z1:（i−1））是（z1:（i−1））中非空白标签的序列。RNN-T架构使用audio encoder、label encoder和联合网络参数化（z | x）。编码器是两个神经网络，分别对输入序列和目标输出序列进行编码。以前的工作[14]采用长-短期记忆模型（LSTM）作为编码器，使RNN-T得名。然而，这个框架并不局限于RNN。在本文中，我们特别感兴趣的是用transformer取代LSTM编码器[1，2]。在下文中，我们将这种新结构称为transformer-transducer（T-T）。与原始RNN-T模型一样，联合网络使用带有softmax层的前馈神经网络，将音频编码器ti时刻的输出，和给定先前非空白输出标签序列标签（z1:（i−1））的标签编码器输出相结合。模型P（zi | x，ti，Labels（z1:（i−1）））定义如下：





其中每个线性函数是不同的单层前馈神经网络，AudioEncodert *ti*（x）是当时的音频编码器输出，LabelEncoder（Labels（z1:（i−1）））是给定先前非空白标签序列的标签编码器输出。

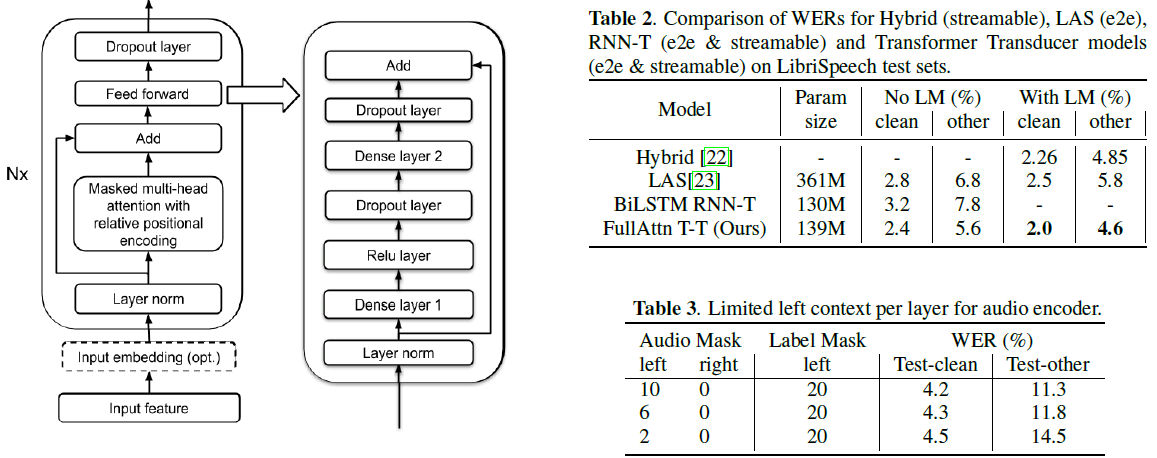
通过简单地求所有有效对齐的和来计算公式（1）在计算上是困难的。因此，我们将前向变量*α*（t，u）定义为所有路径在时间帧*t*和标签位置*u*结束的概率之和。然后，我们使用前向算法[14，21]来计算最后一个alpha变量*α*（T，U），它对应于等式（1）中定义的P（y | x）。使用前向算法的P（y | x）的有效计算是通过以下事实实现的：在任何给定标签位置和任何给定时间帧的局部概率估计（等式（4））不依赖于对齐[14]。然后，模型的训练损失是等式（1）中定义的所有训练示例的负对数概率之和，



其中*Ti*和*Ui*分别是第i训练示例的输入序列和输出目标标签序列的长度。

## 2.2.    transformer

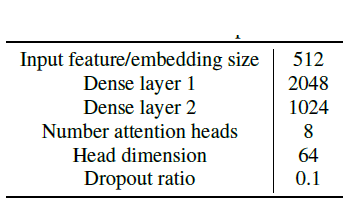
transformer[1]由多个相同层的堆栈组成。每一层有两个子层，一个多头注意层和一个前馈层。我们的多头注意力层首先应用LayerNorm，然后将输入投射到所有head的Query、Key和Value中[2]。注意机制分别适用于不同的注意头。注意机制提供了一种灵活的方式来控制模型使用的上下文。例如，我们可以屏蔽当前帧左侧的注意分数，以产生仅基于先前状态历史的输出。将所有head的加权平均值串联起来，并传递给前馈层。然后，我们在规范化的输入和前馈层的输出上使用残差连接来形成多头注意子层的最终输出（即LayerNorm（x）+AttentionLayer（LayerNorm（x）），其中是多头注意子层的输入）。我们还对前馈层的输出应用了dropout，以防止过拟合。我们的前馈子层首先对输入应用LayerNorm，然后应用两个全连接层。我们用ReLu作为第一个全连接层的激活函数。再次，对这两个全连接层进行正则化，并且应用归一化输入和第二个全连接层（即LayerNorm（x）+前馈层（LayerNorm（x）），其中是对前馈子层的输入）的输出的残余连接。更多细节请参见图2。



请注意，与[1]中的体系结构不同，LabelEncoder状态并不涉及AudioEncoder状态。正如引言中所讨论的，这样做对流应用程序提出了挑战。相反，我们使用上述transformer实现等式（3）中的音频编码器和标签编码器，在传统RNN-T架构[14、16、15]中它们采用LSTM。与上一节描述的RNN-T体系结构相结合，这里的注意机制仅在AudioEncoder或LabelEncoder中运行，这与基于transformer的系统的标准实践相反。此外，为了建立序列顺序模型，我们使用了文献[2]中提出的相对位置编码。对于相对位置编码，编码只影响注意力得分，而不影响被求和的值。这允许我们重用先前计算的状态，当AudioEncoder或LabelEncoder处理的帧或标签的数量大于训练期间使用的最大长度时（这对于流应用程序来说同样是困难的），而不是重新计算所有以前的状态并以重复推理的方式获取最后的状态。更具体地说，运行一步推理以在时间*t*获得激活的复杂度是O（t），这是在使用相对位置编码时处理当前步骤的状态和前馈过程的计算成本。另一方面，对于绝对位置编码，当编码长度*t*大于训练期间使用的最大长度时，添加到输入的编码应移位1，这排除了状态的重复使用，并且使得复杂性为O（t2）。然而，即使我们可以通过相对位置编码将复杂性从O（t2）降低到O（t），仍然存在延迟随时间增长的问题。一个直观的解决方案是将模型限制在一个移动的状态窗口*W*中，使一步推理的复杂性保持不变。请注意，对于从解码器到编码器都有注意力的基于transformer的模型，在有限的上下文中注意训练或推理是不可能的，因为这样的设置本身正试图学习对齐。相比之下，在RNN-T架构中，AudioEncoder和LabelEncoder的分离，以及对齐由单独的前向-后向过程处理的事实，使得在明确指定的、有限的上下文上进行注意力训练成为可能。

图2所示。transformer编码器结构。

表1。transformer编码器参数设置。



# 3.    实验和结果

## 3.1.    数据

我们使用公开的LibriSpeech ASR语料库[24]评估了提出的模型。LibriSpeech数据集包含970个小时的音频数据和相应的文本转录（约10M单词标记）以及一个额外的800M单词标记纯文本数据集。配对音频/转录数据集用于训练T-T模型和基于LSTM的基线。完整的810M word tokens文本数据集用于独立语言模型（LM）训练。我们从一个32ms的窗口中提取128个通道的logmel能量值，每4帧叠加一次，每3帧再采样一次，生成一个512维的声学特征向量，步幅为30ms。在模型训练过程中应用了特征增强[23]，以防止过拟合，提高泛化能力，只有频率掩蔽（F=50，mF=2）和时间掩蔽（T=30，mT=10）。

## 3.2.    transformer transducer

我们的transformer-transducer模型架构有18个音频编码器层和2个标签编码器层。音频和标签编码器的每一层都是相同的。层中计算的详细信息如图2和表1所示。本文提出的所有实验模型都是在8x8 TPU上训练的，每芯batch为16（有效批量为2048）。在最初的4K步中，学习速率从0线性上升到2.5e−4，然后保持不变直到30K步，然后指数衰减到2.5e−6直到200K步。在训练期间，从10K步开始，我们还添加了高斯噪声（*µ*=0，σ=0.01）到模型权重[25]。在我们所有的实验中，我们训练这个模型来输出grapheme unit。我们发现transformer-tranducer模型的训练速度比同参数量的LSTM RNN-T模型快得多（≈1天 vs ≈3.5天）。

## 3.3.    结果

我们首先比较了音频上full attention的transformer-transducer（TT）模型和使用双向LSTM音频编码器的RNN-T模型的性能。如表2所示，T-T模型显著优于基于LSTM的RNN-T基线。我们还观察到，T-T模型可以达到与现有的同等大小的wordpiece-based的端到端模型的竞争识别精度。为了与单独训练LM，进行浅层融合（shallow fusion）[19，26]的系统进行比较，我们还训练了一个基于Transformer的LM，其结构与T-T中使用的标签编码器相同，使用了完整的810M字令牌数据集。这个transformer LM（6层；57M参数）在dev-clean上的困惑度为2.49；使用dropout和更大的模型并没有改善困惑度或WER。然后使用LM和训练的T-T系统以及训练的基于双向LSTM的RNN-T基线执行浅层融合，在LibriSpeech dev数据集上，用缩放因子对LM输出和非空白符号序列长度进行微调。结果如表2“With LM”一栏所示。T-T系统的浅层融合结果与现有最佳系统的相应结果具有竞争性。

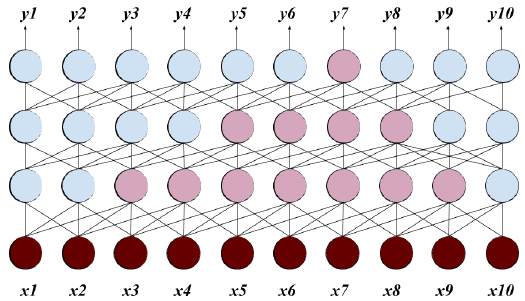


图3。Transformer context masking for y7 position（left=2，right=1）

表4。音频编码器每层的有限右上下文。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| left | right | left | Test-clean | Test-other |
| 512 | 512 | 20 | 2.4 | 5.6 |
| 512 | 10 | 20 | 2.7 | 6.6 |
| 512 | 6 | 20 | 2.8 | 6.9 |
| 512 | 2 | 20 | 3.0 | 7.7 |
| 10 | 0 | 20 | 4.2 | 11.3 |

接下来，我们使用有限窗口attention的T-T模型对音频和文本进行训练和解码实验，以期构建具有低延迟的在线流式语音识别系统。与在线模型中使用单向RNN音频编码器类似，在在线模型中，时间*t*的激活仅在时刻t之前的音频帧上计算，在此，我们通过屏蔽当前帧右侧的注意分数来约束音频编码器只关注当前帧的左侧。为了使AudioEncoder的一步推理具有可处理性（即具有恒定的时间复杂度），我们通过再次掩蔽注意分数，进一步将AudioEncoder的注意限制在先前状态的固定窗口内。由于计算资源有限，我们对不同的transformer层使用相同的掩码，但是对不同的层使用不同的上下文（掩码）是值得探索的。结果如表3所示，其中前两列中的N表示模型在当前帧的左侧或右侧使用的状态数。正如我们所看到的，使用更多的音频历史可以得到更低的WER，但是考虑到一个具有合理时间复杂度的流化模型来进行推理，我们用每层最多10帧的左上下文进行了实验。

表5。标签编码器每层的左上下文有限。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| left | right | Label Mask left | Test-clean | Test-other |
| 10 | 0 | 20 | 4.2 | 11.3 |
| 10 | 0 | 4 | 4.2 | 11.4 |
| 10 | 0 | 3 | 4.2 | 11.4 |
| 10 | 0 | 2 | 4.3 | 11.5 |
| 10 | 0 | 1 | 4.4 | 12 |

表6。限制流媒体的音频和标签上下文的结果。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| left | right | Lm letf | Test-clean | Test-other |
| 512 | 512 | 20 | 2.4 | 5.6 |
| 10 | 2 | 2 | 3.6 | 10 |
| 10 | 0 | 20 | 4.2 | 11.3 |

类似地，我们探索了使用有限的右上下文来允许模型看到一些未来的音频帧，希望能够弥合可流式T-T模型（左=10，右=0）和完全注意T-T模型（左=512，右=512）之间的差距。因为我们对每一层应用相同的掩码，所以通过使用右边的上下文引入的延迟在所有层上累积。例如，在图3中，要从一个具有一帧右上下文的三层transformer生成，它实际上需要等待x10到达，在我们的示例中是90毫秒延迟。为了探索右窗口对建模的影响，我们对每层固定512帧的左窗口与完全注意T-T模型进行了比较。从表4可以看出，对于每层6帧的右上下文（大约3.2秒的延迟），性能比完全注意模型差16%左右。与可流式的T-T模型相比，每层2帧右上下文（大约1秒的延迟）带来大约30%的改进。

此外，我们还评估了T-T LabelEncoder中使用的左上下文如何影响性能。在表5中，我们展示了约束每一层仅使用三个先前的标签状态与每层使用20个状态的模型具有相似的精度。它显示对于T-T模型，非常有限的左上下文标签编码器就能达到很好的结果。当使用full attention的T-T音频编码器，限制左标签状态时，我们看到了类似的趋势。

最后，表6报告了使用10帧的有限左上下文时的结果，这将一步推理的时间复杂度降低到一个常数，并对未来帧进行展望，以此弥补仅左注意模型和完全注意模型之间的差距。

# 4.    结论

在本文中，我们提出了transformer-transducer模型，在RNN-T架构中嵌入基于transformer的音频自注意和标签编码，产生一个端到端模型，该模型可以使用一个损耗函数进行优化，该损耗函数可以有效地统计所有可能的对齐，并且非常适合时间同步解码。该模型在LibriSpeech基准上获得了最新的准确率，并且通过限制用于自我注意的音频和标签上下文，可以方便地用于流式语音识别。transformer-transducer模型的训练速度明显快于基于LSTM的RNN-T模型，它们允许我们以灵活的方式权衡识别精度和延迟。

# 5.    参考文献

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in Advances in neural information processing systems, 2017, pp. 5998–6008.

[2] Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov, “Transformer-xl: Attentive language models beyond a fixed-length context,” in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, p. 29782988.

[3] Kazuki Irie, Albert Zeyer, Ralf Schlter, and Hermann Ney, “Language Modeling with Deep Transformers,” in Proc. Interspeech, 2019, pp. 3905–3909.

[4] Linhao Dong, Shuang Xu, and Bo Xu, “Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition,” in Proc. ICASSP, 04 2018, pp. 5884–5888.

[5] Ngoc-Quan Pham, Thai-Son Nguyen, Jan Niehues, Markus M¨uller, and Alex Waibel, “Very deep self-attention networks for end-to-end speech recognition,” CoRR, vol. abs/1904.13377, 2019.

[6] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F.Wong, and Lidia S. Chao, “Learning deep transformer models for machine translation,” CoRR, vol. abs/1906.01787, 2019.

[7] “Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin chinese,” in Proc. Interspeech 2018. pp. 791–795, ISCA.

[8] Abdelrahman Mohamed, Dmytro Okhonko, and Luke Zettlemoyer, “Transformers with convolutional context for ASR,” CoRR, vol. abs/1904.11660, 2019.

[9] Daniel Povey, Hossein Hadian, Pegah Ghahremani, Ke Li, and Sanjeev Khudanpur, “A time-restricted self-attention layer for asr,” 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5874–5878, 2018.

[10] Linhao Dong, Feng Wang, and Bo Xu, “Self-attention aligner: A latency-control end-to-end model for asr using self-attention network and chunk-hopping,” 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), May 2019.

[11] Matthias Sperber, Jan Niehues, Graham Neubig, Sebastian Stker, and Alex Waibel, “Self-attentional acoustic models,” Proc. Interspeech, Sep 2018.

[12] Emiru Tsunoo, Yosuke Kashiwagi, Toshiyuki Kumakura, and Shinji Watanabe, “Towards online end-to-end transformer automatic speech recognition,” arXiv:1910.11871, 2019.

[13] Has¸im Sak, Andrew Senior, and Francoise Beaufays, “Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling,” in Proc. Interspeech, 2014.

[14] Alex Graves, “Sequence transduction with recurrent neural networks,” in Proceedings of the 29th International Conference on Machine Learning, 2012.

[15] Kanishka Rao, Has¸im Sak, and Rohit Prabhavalkar, “Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer,” in 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2017, pp. 193–199.

[16] Yanzhang (Ryan) He, Rohit Prabhavalkar, Kanishka Rao, Wei Li, Anton Bakhtin, and Ian McGraw, “Streaming smallfootprint keyword spotting using sequence-to-sequence models,” in Automatic Speech Recognition and Understanding (ASRU), 2017 IEEE Workshop on, 2017.

[17] Has¸im Sak, Matt Shannon, Kanishka Rao, and Franoise Beaufays, “Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping,” in Proc. Interspeech, 2017, pp. 1298–1302.

[18] Navdeep Jaitly, David Sussillo, Quoc V Le, Oriol Vinyals, Ilya Sutskever, and Samy Bengio, “A neural transducer,” arXiv preprint arXiv:1511.04868, 2015.

[19] Alex Graves, Santiago Fern´andez, Faustino Gomez, and J¨urgen Schmidhuber, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks,” in Proc. ICML, 2006.

[20] Anshuman Tripathi, Han Lu, Hasim Sak, and Hagen Soltau, “Monotonic Recurrent Neural Network Transducer and Decoding Strategies,” in Proc. ASRU, 2019.

[21] L. R. Rabiner and B.-H. Juang, Fundamentals of Speech Recognition, PTR Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1993.

[22] Yongqiang Wang, Abdelrahman Mohamed, Duc Le, Chunxi Liu, Alex Xiao, Jay Mahadeokar, Hongzhao Huang, Andros Tjandra, Xiaohui Zhang, Frank Zhang, Christian Fuegen, Geoffrey Zweig, and Michael L. Seltzer, “Transformerbased acoustic modeling for hybrid speech recognition,” arXiv:1910.09799, 2019.

[23] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, “Specaugment: A simple data augmentation method for automatic speech recognition,” arXiv preprint arXiv:1904.08779, 2019.

[24] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in Proc. ICASSP. IEEE, 2015, pp. 5206– 5210.

[25] Alex Graves, “Practical variational inference for neural networks,” in Advances in neural information processing systems, 2011, pp. 2348–2356.

[26] Jan Chorowski and Navdeep Jaitly, “Towards better decoding and language model integration in sequence to sequence models,” in Proc. Interspeech, 2017, pp. 523–527.