

Listen, Attend and Spell

听、注意和拼写



|  |  |
| --- | --- |
| 陈冯富珍  卡内基梅隆大学邮箱：williamchan@cmu.edu | 纳夫迪普·贾特利、库克诉勒、奥里尔·维尼亚  谷歌大脑  {ndjaitly，qvl，vinyals}@谷歌网站 |

摘要

We present Listen, Attend and Spell (LAS), a neural network that learns to transcribe speech utterances to characters. Unlike traditional DNN-HMM models, this model learns all the components of a speech recognizer jointly. Our system has two components: a listener and a speller. The listener is a pyramidal recurrent network encoder that accepts filter bank spectra as inputs. The speller is an attentionbased recurrent network decoder that emits characters as outputs. The network produces character sequences without making any independence assumptions between the characters. This is the key improvement of LAS over previous end-toend CTC models. On a subset of the Google voice search task, LAS achieves a word error rate (WER) of 14.1% without a dictionary or a language model, and 10.3% with language model rescoring over the top 32 beams. By comparison, the state-of-the-art CLDNN-HMM model achieves a WER of 8.0%.

我们提出了听，参加和拼写（LAS），一个神经网络，学习转录语音文字。与传统的DNN-HMM模型不同，该模型联合学习语音识别器的所有组成部分。我们的系统有两个组件：侦听器和拼写器。监听器是一个金字塔结构的递归网络编码器，它接受滤波器组谱作为输入。拼写器是一个基于注意的循环网络解码器，它以字符作为输出。网络生成字符序列时，不需要对字符之间的独立性作任何假设。这是LAS相对于以前的端到端CTC模型的关键改进。在googlevoice搜索任务的一个子集上，LAS在没有词典或语言模型的情况下实现了14.1%的单词错误率（WER），在前32个波束上进行语言模型重扫描时达到了10.3%。相比之下，最先进的CLDNN-HMM模型的加权率达到8.0%。

2015

# 1        介绍

深度神经网络（DNNs）使得语音识别器的各个组成部分得到了改进。它们通常用于混合DNN-HMM语音识别系统的声学建模[1，2，3，4，5，6]。DNN在将单词映射到音素序列的发音模型中也取得了显著的进展[7，8]。在语言建模中，递归模型被证明可以通过重新选择n个最佳列表来提高语音识别的准确性[9]。传统上，这些组成部分——声学、发音和语言模型——都是单独训练的，每个都有不同的目标。最近在这一领域的工作试图通过设计端到端训练的模型来纠正这种不相交的训练问题——从语音直接训练到转录[10，11，12，13，14，15]。这方面的两种主要方法是连接主义时间分类（CTC）[10]和带注意的序列到序列模型[16]。这两种方法都有我们试图解决的局限性：CTC假设标签输出在条件上相互独立；而序列到序列的方法仅适用于音素序列[14，15]，而没有经过端到端的语音识别训练。

本文介绍了Listen，attent和Spell（LAS），这是一种神经网络，它改进了以前的尝试[12，14，15]。网络学习将音频序列信号转录成单词序列，一次一个字符。与以前的方法不同，LAS不在标签序列中进行独立性假设，也不依赖HMMs。LAS是基于顺序对顺序的学习框架和注意[17，18，16，14，15]。它由一个编码器递归神经网络（RNN）和一个解码器RNN组成，前者称为侦听器，后者称为拼写器。监听器是一个金字塔状的RNN，它将低级语音信号转换成高级特征。拼写器是一种RNN，它通过使用注意机制指定字符序列的概率分布，将这些更高层次的特征转换为输出话语[16，14，15]。听者和拼写者是联合训练的。

我们的方法的关键在于，我们为听者使用金字塔RNN模型，这减少了注意力模型必须从中提取相关信息的时间步数。由于模型每次输出一个字符的字符序列，罕见和词汇表外（OOV）单词将自动处理。将字符建模为输出的另一个优点是网络能够自然地生成多个拼写变体。例如，对于短语“3A”，模型在顶梁中产生“3A”和“aaa”（见第4.5节）。像CTC这样的模型可能会因为框架之间的条件独立假设而难以为同一个话语产生如此不同的转录。

在我们的实验中，我们发现这些组件是LAS正常工作所必需的。在没有注意机制的情况下，该模型对训练数据的拟合程度明显过高，尽管我们的训练集有300万个话语——它在不注意声学的情况下记忆训练记录。如果编码器端没有金字塔结构，我们的模型收敛太慢——即使经过一个月的训练，错误率也明显高于我们在这里报告的错误。这两个问题的产生都是因为声音信号可能有数百到数千帧，这使得训练RNN变得困难。最后，为了减少拼写者对训练记录的过度拟合，我们在训练期间使用了抽样技巧[19]。

通过这些改进，LAS在Google语音搜索任务的一个子集上获得了14.1%的WER，无需词典或语言模型。当与语言模型重排序相结合时，LAS达到了10.3%的WER。相比之下，谷歌最先进的CLDNN-HMM系统在相同的数据集上实现了8.0%的功耗[20]。

# 2        相关工作

尽管深度网络已成功应用于许多应用，但直到最近，它们主要用于分类：将固定长度向量映射到输出类别[21]。对于结构化问题，例如将一个变长序列映射到另一个变长序列，神经网络必须与其他序列模型相结合，如隐马尔可夫模型（HMM）[22]和条件随机场（CRF）[23]。这种组合方法的一个缺点是，生成的模型不能很容易地进行端到端的训练，并且它们对数据的概率分布作出了过于简单的假设。

序列到序列学习是一个试图解决学习可变长度输入和输出序列问题的框架[17]。它使用编码器RNN将连续可变长度输入映射为固定长度向量。解码器RNN然后使用该向量产生可变长度的输出序列，一次一个令牌。在训练过程中，模型将groundtruth标签作为输入提供给解码器。在推理过程中，模型进行波束搜索，为下一步预测生成合适的候选。

序列到序列模型可以通过使用注意机制得到显著改进，注意机制在解码器产生输出令牌时为其提供更多信息[16]。在每个输出步骤，解码器RNN的最后隐藏状态用于在编码器的输入序列上生成注意向量。注意向量用于在每个时间步将信息从编码器传播到解码器，而不是像原始序列到序列模型那样只传播一次[17]。这个注意向量可以被认为是跳跃连接，允许信息和梯度在RNN中更有效地流动。

序列到序列框架已被广泛应用于许多应用领域：机器翻译[24，25]、图像字幕[26，27]、解析[28]和会话建模[29]。这个框架的普遍性表明，语音识别也可以是一个直接的应用[14，15]。

# 3        模型

在本节中，我们将正式描述接受声学特征作为输入并发射英语字符作为输出的LAS。设=（x1，…，xT）为滤波器组光谱特征的输入序列，设=（hsosi，y1，…，yS，heosi），∈{a，b，c，···，z，0，···，9，hspacei，hcommai，hperiodi，hapostrophei，hunki}，为字符的输出序列。这里hsosi和heosi分别是特殊的句首标记和句尾标记。**x和y***彝语*

我们希望使用链式规则将每个字符输出建模为前一个字符和输入信号的条件分布：*彝语y<i***十**

*第*（yx |）=是的*第*（yi | x，y<i）(1)

*我*

我们的Listen，attent和Spell（LAS）模型由两个子模块组成：listener和speller。侦听器是一个声学模型编码器，其关键操作是侦听。拼写器是一个基于注意的字符解码器，其关键操作是AttendAndSpell。Listen函数将原始信号转换为≤T的高级表示=（h1，…，hU），而AttendAndSpell函数消耗并产生字符序列的概率分布：**x高h高***U型*

**小时**=听（x）(2)

*第*（yxhy |）=AttendAndSpell（，）(3)

图1显示了具有这两个组件的LAS。我们将在以下部分中提供这些组件的更多详细信息。

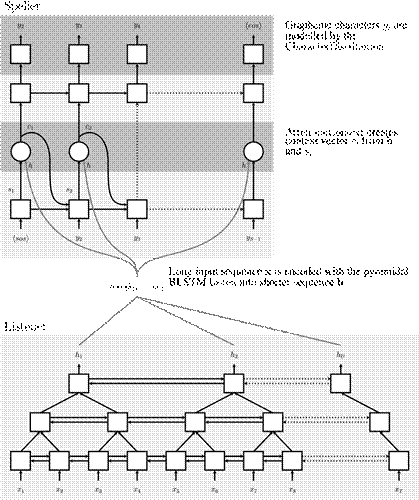


图1：Listen，attent和Spell（LAS）模型：listener是一个金字塔结构的BLSTM，它将我们的输入序列编码为高级特征，speller是一个基于注意的解码器，从中生成字符。**x高**

3.1倾听

侦听操作使用具有金字塔结构的双向长-短期存储器RNN（BLSTM）[30，31，12]。由于输入语音信号可以是数百到数千帧长，因此需要进行这种修改以减少输入的长度。BLSTM在Listen手术中的直接应用收敛缓慢，即使经过一个月的训练，结果也不如本文报道的结果。这大概是因为操作很难从大量的输入时间步中提取相关信息。*U型***hx公司***T型*

我们通过使用类似于发条RNN的金字塔BLSTM（pBLSTM）来规避这个问题[33]。在每个连续叠加的pBLSTM层中，我们将时间分辨率降低了2倍。在典型的深BTLM体系结构中，第-层在第-时间步的输出计算如下：*我日本*

*hji公司*=BLSTM((4)

在pBLSTM模型中，我们将每一层的连续步骤的输出串联起来，然后将其馈送到下一层，即：

=pBLSTM((5)

在我们的模型中，我们将3个PBLSTM堆叠在底部BLSTM层的顶部，以降低时间分辨率23=8倍。这使得注意力模型（见下一节）能够从较少的步骤中提取相关信息。除了降低分辨率之外，deep架构还允许模型学习数据的非线性特征表示。pBLSTM的可视化如图1所示。

金字塔结构也降低了计算复杂度。在下一节中，我们将说明特征上的注意机制的计算复杂性为（US）。因此，减少学习和推理速度大大加快。*U型OU型*

3.2参加并拼写

现在我们来描述AttendAndSpell函数。使用基于注意的LSTM传感器计算该函数[16，15]。在每个输出步骤中，传感器根据前面看到的所有字符生成下一个字符的概率分布。的分布是解码器状态和上下文的函数。解码器状态是先前状态、先前发射的字符和上下文的函数。上下文向量是由注意机制产生的。明确地，*彝语硅ci公司硅硅*−1*彝语*−1*ci公司*−1*ci公司*

|  |  |
| --- | --- |
| *ci公司*=注意上下文（si，h） | (6) |
| *硅*=RNN（硅负极）1*，彝语*1*，ci公司*1 | (7) |

*第*（yi | x，y<i）=字符分布（s，cii）(8)

其中CharacterDistribution是一个MLP，softmax输出超过个字符，RNN是一个2层LSTM。

在每个时间步，注意机制AttentionContext生成一个上下文向量，将生成下一个字符所需的信息封装在声学信号中。注意模型是基于内容的-解码器状态的内容与表示时间步长的内容匹配，以生成注意向量。用于线性混合向量以创建。*我ci公司硅胡u型***小时***αiαi胡ci公司*

具体地说，在每个解码器时间步，AttentionContext函数使用向量∈h和向量计算每个时间步的标量能量。使用softmax函数将标量能量转换为步长（或注意）的概率分布。这用于通过在不同的时间步线性混合侦听器功能来创建上下文向量：*我ei，美国u型胡硅ei，美国αici公司胡*

*ei，美国*=hφ（si），ψ（hu）i(9)

(10)

(11)

MLP网络的位置和位置。在收敛性上，分布通常是非常尖锐的，并且只集中在少数帧上；可以看作是一个连续的加权特征包。图1显示了LAS体系结构。*φψαi***啊***ci公司*

3.3学习

在端到端语音识别中，可以对听音和拼写功能进行联合训练。序列到序列的方法将下一步预测置于前面的字符[17，16]上，并使对数概率最大化：

最大对数p（yi | x，y<\*i；θ）（12）十*θ*

*我*

前几个角色的真实性在哪里。*是的<*∗*我*

然而，在推理过程中，由于没有训练模型在某些时间步长上对错误的预测具有适应能力，因此缺少基本事实，预测可能会受到影响。为了改善这种效果，我们使用了[19]中提出的技巧。在训练过程中，我们有时会从之前的字符分布中取样，并将其作为下一步预测的输入，而不是总是输入地面真相转录本进行下一步预测：

|  |  |
| --- | --- |
| *是的*∼i∼字符分布（si，ci） | (13) |
| 最大对数p（yi | x，y‘<i；θ）十 | (14) |

*θ*

*我*

式中，i−1是从基本真值中选择的字符，或从具有特定采样率的模型中采样的字符。与[19]不同的是，我们不使用时间表，只是从训练开始就使用10%的恒定采样率。*是的*

由于该系统是一个非常深入的网络，因此可能需要某种类型的预培训。然而，在我们的实验中，我们发现没有必要进行预训练。特别地，我们尝试用传统的GMM-HMM系统生成的上下文无关或上下文相关的音素来预训练Listen函数。一个softmax网络连接到监听器的输出单元∈h上，用于进行多帧音素状态预测[34]，但没有改进。我们也尝试将这些音素作为联合目标[35]，但没有发现任何改进。*胡*

3.4解码和重排序

在推断过程中，我们希望找到给定输入声学的最可能的字符序列：

**是的**ˆ=argmaxlogP（yx |）(15)

**是的**

解码是用类似于[17]的简单的从左到右波束搜索算法执行的。我们维持了一系列的部分假设，从句子的开头开始。在每一个时间步，波束中的每一部分假设都用每一个可能的特征展开，并且只保留最可能的波束。当遇到heosi标记时，它将从波束中移除并添加到完整假设集。可以选择添加字典来将搜索空间限制为有效单词，但是我们发现这不是必需的，因为模型几乎一直在学习拼写真实单词。*ββ*

我们有大量的文本数据[36]，与转录的语音相比。

我们可以使用单独在文本语料库上训练的语言模型，类似于传统的语音系统[37]。为此，我们可以用语言模型重新扫描光束。我们发现我们的模型对较短的话语有很小的偏差，因此我们通过假设中的字符数| y | c来规范我们的概率，并将其与语言模型概率（y）结合起来：*第*流光溢彩

                                                      *s码*（y）流光溢彩（年）(16)

*c级*

其中是我们的语言模型权重，可以通过一个保留的验证集来确定。*λ*

# 4        实验

我们在实验中使用了大约300万个谷歌语音搜索话语（代表2000小时的数据）的数据集。大约10小时的话语被随机选择作为一个保持验证集。使用室内模拟器进行数据增强，添加不同类型的噪声和混响；噪声源来自YouTube和日常事件的环境记录[20]。这使音频数据量增加了20倍。每10ms计算40维log-mel滤波器组特征，并将其作为听者的声学输入。一组单独的22K的话语代表大约16小时的数据被用作测试数据。还使用应用于训练数据的相同损坏策略创建了噪声测试数据集。所有的训练集都是匿名和手写的，并且是google语音流量的代表。

文本通过将所有字符转换为小写英文字母数字（包括数字）进行规范化。标点符号：空格、逗号、句点和撇号被保留，而所有其他标记被转换成未知的hunki标记。如前所述，所有的话语都用句子开头的hsosi和句子结尾的heosi标记填充。

这个数据集上最先进的模型是一个CLDNN-HMM系统，如[20]所述。CLDNN系统在干净的测试集上达到了8.0%的功率，在有噪声的测试集上达到了8.9%。然而，我们注意到CLDNN使用单向cldns，并且肯定会从双向CLDNN体系结构的使用中受益更多。

对于Listen函数，我们在操作输入的BLSTM上使用了3层512个pBLSTM节点（即每个方向256个节点）。这将时间分辨率降低了8=23倍。Spell函数使用了一个两层的LSTM，每个LSTM有512个节点。使用均匀分布U（−0.1,0.1）初始化权重。

异步随机梯度下降（ASGD）用于训练我们的模型[38]。学习率为0.2，几何衰减率为每3M话语0.98次（即一个时代的第20次）。我们使用了distbelieve框架[38]，其中有32个副本，每个副本都有一个小批量的32个语句。为了进一步加快训练速度，根据帧长将序列分组到桶中[17]。1*/*

在验证集的结果停止改善之前，使用groundtruth先前的字符对模型进行训练。这花了大约两周的时间。在没有任何字典或语言模型的情况下，使用波束宽度=32对模型进行解码，在干净的测试集上实现了16.2%的功耗，在有噪声的测试集上实现了19.0%的功耗。我们发现用字典约束beam搜索对WER没有影响。使用CLDNN系统使用的相同n-gram语言模型（语言模型权重=0.008）重新扫描前32个梁，将干净和噪声测试集的结果分别提高到12.6%和14.7%。请注意，为了方便起见，我们没有使用语言模型进行解码，而是只对前32个波束进行了重新扫描。在解码期间使用语言模型有可能获得进一步的增益。*βλ*

如第3.3节所述，培训和测试之间存在不匹配。在训练过程中，模型以正确的先前字符为条件，但在测试过程中，模型所犯的错误会破坏未来的预测。我们训练了另一个模型，从之前的字符分布中抽样，概率为10%（我们没有使用[19]中描述的时间表）。在没有使用语言模型重排序的情况下，这将干净和有噪声测试集的结果分别提高到14.1%和16.5%。通过语言模型重构，我们在干净和有噪声的测试集上分别获得了10.3%和12.0%的信噪比。表1总结了这些结果。

在干净的测试集上，该模型的绝对功率在最先进的CLDNN-HMM系统的2.5%以内，而在有噪声的测试集上，该模型的绝对功率小于3.0%。我们怀疑-

表1：干净和嘈杂的Google语音搜索任务的WER比较。CLDNN-HMM系统是最先进的系统，侦听、参与和拼写（LAS）模型的解码波束大小为32。语言模型（LM）重扫描被应用到我们的梁，并采用抽样技巧来弥补训练和推理之间的差距。

|  |  |  |
| --- | --- | --- |
| 模型 | 清洁能源 | 噪音 |
| CLDNN-HMM[20] | 8.0 | 8.9 |
| 拉斯维加斯 | 16.2 | 19.0 |
| LAS+LM重新扫描 | 12.6 | 14.7 |
| LAS+取样 | 14.1 | 16.5 |
| LAS+采样+LM重新扫描 | 10.3 | 12.0 |

与非卷积结构相比，非卷积结构滤波器在干净语音和噪声语音上的性能分别提高了5%和7%。

4.1注意力可视化

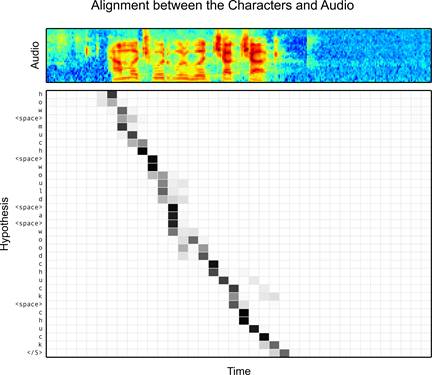


图2:Listen，attent和Spell（LAS）模型产生的字符输出和音频信号之间的对齐“一只土拨鼠要多少钱”。基于内容的注意机制能够正确识别第一个字符在音频序列中的起始位置。产生的对齐通常是单调的，不需要任何基于位置的先验。

基于内容的注意机制在字符和音频信号之间建立了明确的对齐。我们可以通过记录每个字符输出时间步长上的注意分布来可视化注意机制。图2显示了“一只土拨鼠要多少钱”这句话的字符和过滤器组之间的注意力对齐。对于这个特殊的话语，模型学习了一个没有任何位置先验的单调分布。“土拨鼠”和“chuck”在声学上有相似之处，当“土拨鼠”在分布上被稀释时，注意机制稍显混乱。注意模型也能正确识别话语的开始和结束。

在下面的部分中，我们报告了控制实验的结果，这些实验是为了理解波束宽度、话语长度和词频对我们模型的WER的影响。

4.2波束宽度的影响

我们研究了模型性能与波束搜索宽度之间的相关性，以及语言模型重定和不重定的情况。图3显示了解码波束宽度对干净测试集的WER的影响。我们通过将光束宽度增加到16来观察到持续的WER改善，之后我们观察到没有明显的好处。在波束宽度为32的情况下，经过语言模型重构后的WER分别为14.1%和10.3%。用oracle重新扫描前32个梁，在干净的测试集上产生4.3%的WER，在有噪声的测试集上产生5.5%的WER。*β*

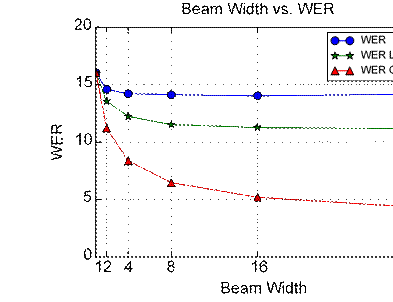


图3：解码波束宽度对清除Google语音搜索任务的WER的影响。所报道的WER没有字典或语言模型，语言模型重新扫描和不同波束宽度的oracle WER。该图表明，即使光束尺寸相对较小，也可以获得良好的结果。

4.3话语长度的影响

我们将模型的性能作为话语中字数的函数来衡量。由于在我们的分布中长训练话语的数量有限，我们期望该模型在长话语上做得很差。因此，较长的话语有较大的错误率也就不足为奇了。长句子的错误主要是删除，这表明我们可能遗漏了单词。令人惊讶的是，简短的话语（例如，2个单词或更少）表现得相当差。在这里，替换和插入是错误的主要来源，这表明该模型可能会将单词分开。

图4还表明，我们的模型在训练较短话语的分布时，很难推广到长话语。正如文献[15]所报道的，基于位置的先验可能在这些情况下有所帮助。

4.4词频

我们研究了我们的模型在稀有词上的性能。我们使用召回指标来表示一个词是否出现在话语中，而不管其位置如何（越高越好）。图5报告了测试分布中每个单词的回忆，作为训练分布中单词频率的函数。稀有词具有较高的方差和较低的回忆，而更频繁的词通常具有较高的记忆

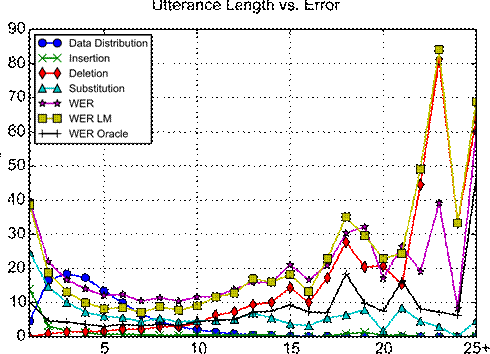


图4：错误率（插入、删除、替换和WER）与话语中单词数量之间的关系。WER报告没有字典或语言模型，语言模型重新排序，oracle WER用于干净的Google语音搜索任务。图中覆盖了与一句话的字数有关的数据分布。尽管有大量的数据，LAS在短句中的表现很差。LAS也不能很好地概括较长的话语时，训练的分布较短的话语。插入和替换是短句错误的主要来源，而删除是长句错误的主要来源。

回忆。“and”这个词在训练集中出现了85k次，但是即使在重新调整语言模型后，它的召回率也只有80%。“和”这个词经常被误译为“in”（有95%的回忆）。这表明语言模型需要改进。相比之下，“walkerville”这个词在训练集中只出现一次，但它的召回率是100%。这表明对一个单词的回忆既取决于它在训练集中的频率，也取决于它在声学上的独特性。

4.5有趣的解码示例

在本节中，我们将展示该模型在多个语句上的输出，以展示LAS的功能。本节中的所有结果都是在没有词典或语言模型的情况下解码的。

在我们的实验中，我们观察到，在相同的声学条件下，LAS可以学习多种拼写变体。表2显示了包含“3A”的语句的顶梁。可以看出，该模型在前四个梁中同时产生“3A”和“aaa”。由于下一步预测模型利用链式规则分解对概率分布不作任何假设，解码器能够产生这种变化的解析。由于条件独立性假设，（yi | x）在条件上独立于（yi+1 | x），使用CTC很难产生如此不同的转录本。传统的DNN-HMM系统要求两种拼写都出现在发音词典中，以生成两种拼写排列。*第第*

还可以看到，模型产生了“xxx”，尽管在声学上“x”与“a”有很大的不同——这大概是因为在这种情况下，语言模型压倒了声学信号。在训练语料库中，“xxx”是一个非常常见的短语，我们怀疑它的语言模型

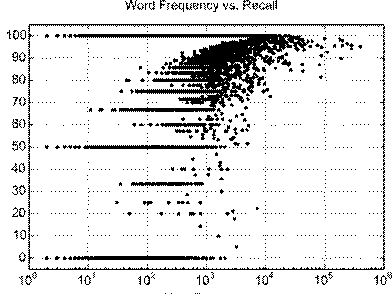


图5：训练分布中的词频和测试分布中的回忆之间的相关性。

一般来说，很少的单词比更频繁的单词更难回忆。

表2：示例1：“3A”与“aaa”拼写变体。

|  |  |  |  |
| --- | --- | --- | --- |
| 光束 | 文本 | 对数概率 | WER公司 |
| 真相 | 呼叫aaa路边救援 | - | - |
| 1 | 呼叫aaa路边救援 | -0.5740 | 0.00 |
| 2 | 打电话给三甲路边救援 | -1.5399 | 50.00 |
| 3 | 呼叫道路救援 | -3.5012 | 50.00 |
| 4 | 呼叫xxx路边救援 | -4.4375 | 25.00 |

暗含在拼写者学会联系“三重”与“xxx”。我们注意到，“aaa”在训练分布中出现4次，“aaa”（当发音为“aaa”而不是“a”-“a”-“a”）在训练分布中只出现一次。

我们还感到惊讶的是，尽管该模型使用了基于内容的注意，但它仍然能够处理具有重复单词的话语。表3显示了一个重复单词的例子。由于LAS实现了基于内容的注意，因此人们期望它在解码步骤中“失去注意”，产生一个单词的次数比说这个单词的次数多或少。从这个例子可以看出，即使“七”被重复了三次，模型还是成功地输出了“七”三次。这暗示对于重复的内容可能不需要基于位置的先验（例如，基于位置的注意或基于位置的正则化）。

表3：例2：重复“七”s。

|  |  |  |  |
| --- | --- | --- | --- |
| 光束 | 文本 | 对数概率 | WER公司 |
| 真相 | 八九四减七七七 | - | - |
| 1 | 八九四减七七七 | -0.2145 | 0.00 |
| 2 | 8949777 | -1.9071 | 14.29 |
| 3 | 八九四减七七七 | -4.7316 | 14.29 |
| 4 | 八九四九七七 | -5.1252 | 28.57 |

# 5        结论

我们提出了一种基于注意的神经网络，它可以将声音信号直接转录到字符中。LAS基于序列到序列的框架，在编码器中采用金字塔结构，减少了解码器必须处理的时间步数。LAS是端到端训练的，有两个主要组件。第一个组件，监听器，是一个金字塔形的声学RNN编码器，它将输入序列转换为高级特征表示。第二个组件是拼写器，它是一个RNN解码器，负责处理高级特征，并一次拼写出一个字符。我们的系统不使用音素的概念，也不依赖发音词典或hmm。我们绕开CTC的条件独立性假设，展示我们如何学习一个隐式语言模型，该模型可以在相同的声学条件下生成多个拼写变体。为了进一步改善结果，我们在解码器中使用来自softmax分类器的样本作为训练期间下一步预测的输入。最后，我们展示了如何使用一个在附加文本上训练的语言模型来重新审视我们的主要假设。

# 致谢

我们感谢Tara Sainath和Babak Damavandi在数据、语言模型和有用的评论方面对我们的帮助。我们还感谢安德鲁·戴、阿希什·阿加瓦尔、萨米·本吉奥、尤金·布雷夫多、格雷格·科拉多、安德鲁·戴、杰夫·迪恩、拉贾特·蒙加、克里斯托弗·奥拉、迈克·舒斯特、诺姆·沙泽尔、伊利亚·萨茨凯、文森特·范霍克和谷歌大脑团队提供的有益意见、建议和技术援助。

# 工具书类

[1] Nathaniel Morgan和Herve Bourlard。基于隐马尔可夫模型的多层感知器连续语音识别。1990年在IEEE声学、语音和信号处理国际会议上。

[2] 阿卜杜勒·拉赫曼·穆罕默德、乔治·E·达尔和杰弗里·E·辛顿。电话识别的深信不疑网络。神经信息处理系统：语音识别和相关应用的深度学习研讨会，2009年。

[3] 乔治·E·达尔、董宇、李登和亚历克斯·阿塞罗。基于上下文相关dbn-hmms的大词汇量连续语音识别。在2011年IEEE声学、语音和信号处理国际会议上。

[4] 阿卜杜勒·拉赫曼·穆罕默德、乔治·E·达尔和杰弗里·辛顿。基于深度信任网络的声学建模。IEEE音频、语音和语言处理学报，20（1）：14–22，2012。

[5] Navdeep Jaitly、Patrick Nguyen、Andrew W.Senior和Vincent Vanhoucke。预训练深度神经网络在大词汇量语音识别中的应用。在INTERSPEECH，2012年。

[6] Tara Sainath、Abdel rahman Mohamed、Brian Kingsbury和Bhuvana Ramabhadran。LVCSR的深度卷积神经网络。在2013年IEEE声学、语音和信号处理国际会议上。

[7] 拉奥、彭富春、萨克和波菲。利用长-短期记忆递归神经网络实现字音转换。2015年IEEE声学、语音和信号处理国际会议。

[8] 姚凯生和杰弗里·茨威格。字音转换的序列到序列神经网络模型。2015

[9] 托马斯·米科洛夫、卡拉菲亚特·马丁、布尔盖特·卢卡、埃尔诺基·扬和胡丹普尔·桑耶夫。基于递归神经网络的语言模型。INTERSPEECH，2010年。

[10] 亚历克斯·格雷夫斯、圣地亚哥·费尔南德斯、福斯蒂诺·戈麦斯和尤尔根·施密杜伯。连接主义时间分类：用递归神经网络标记未分段的序列数据。在2006年国际机器学习会议上。

[11] 亚历克斯·格雷夫斯。用递归神经网络进行序列转导。参加2012年国际机器学习会议：表征学习研讨会。

[12] 亚历克斯·格雷夫斯和纳夫迪普·贾特利。基于递归神经网络的端到端语音识别。2014年国际机器学习会议。

〔13〕Awni Hannun、卡尔案、Jared Casper、Bryan Catanzaro、Greg Diamos、Erich Elsen、Ryan Prenger、Ryan Prenger、Y、Y、和。深度语音：扩展端到端语音识别。在http://arxiv.org/abs/1412.5567, 2014.

[14] 简·乔洛夫斯基、德米特里·巴达瑙、赵京贤和本吉奥。使用基于注意的递归神经网络的端到端连续语音识别：第一个结果。神经信息处理系统：研讨会深度学习和表征学习研讨会，2014年。

[15] 简·乔洛夫斯基、德米特里·巴达瑙、德米特里·塞尔杜克、赵京贤和本吉奥。基于注意的语音识别模型。在http://arxiv.org/abs/1506.07503, 2015.

[16] Dzmitry Bahdanau、Kyunghyun Cho和Yoshua Bengio。神经机器翻译的联合学习对齐和翻译。2015年国际学习代表大会。

[17] 伊利亚·萨茨凯、奥里奥尔·维尼亚和库克·勒。用神经网络进行序列间学习。神经信息处理系统，2014。

[18] 赵京贤、巴特·范·梅里恩波尔、卡格拉·古尔切勒、德米特里·巴达瑙、费提·布加勒斯、霍尔格·施文和约书亚·本吉奥。使用RNN编码器学习统计机器翻译中的短语表示。在2014年自然语言处理经验方法会议上。

[19] 萨米·本吉奥、奥里奥尔·维尼亚、纳夫迪普·贾特利和诺姆·沙泽尔。递归神经网络序列预测的计划抽样。在http://arxiv.org/abs/1506.03099, 2015.

[20] Tara N.Sainath、Oriol Vinyals、Andrew Senior和Hasim Sak。卷积，长期短期记忆，完全连接的深层神经网络。2015年IEEE声学、语音和信号处理国际会议。

[21]Alex Krizhevsky、Ilya Sutskever和Geoffrey E.Hinton。基于深度卷积神经网络的图像分类。神经信息处理系统，2012。

[22]Leonard E.Baum和Ted Petrie。有限状态马尔可夫链概率函数的统计推断。《数理统计年鉴》，37:1554–15631966年。

[23]约翰·拉弗蒂、安德鲁·麦卡勒姆和费尔南多·佩雷拉。条件随机场：用于分割和标记序列数据的概率模型。在2001年国际机器学习会议上。

[24]Minh Thang Luong、Ilya Sutskever、Quoc V.Le、Oriol Vinyals和Wojciech Zaremba。解决神经机器翻译中的稀有字问题。计算语言学协会，2015年。

[25]Sebastien Jean、Kyunghyun Cho、Roland Memisevic和Yoshua Bengio。使用非常大的目标词汇进行神经机器翻译。计算语言学协会，2015年。

[26]奥利奥·维尼亚（Oriol Vinyals）、亚历山大·托舍夫（Alexander Toshev）、萨米·本吉奥（Samy Bengio）和杜米特鲁·二汉（Dumitru Erhan）。展示和讲述：一个神经图像发生器。在2015年IEEE计算机视觉和模式识别会议上。

[27]Kelvin Xu、Jimmy Ba、Ryan Kiros、Kyunghyun Cho、Aaron Courville、Ruslan Salakhutdinov、Richard Zemel和Yoshua Bengio。展示、参与和讲述：视觉注意力的神经图像字幕生成。在2015年国际机器学习会议上。

[28]奥里奥尔·维尼亚（Oriol Vinyals）、卢卡斯·凯撒（Lukasz Kaiser）、特里·辜胜阻（Terry Koo）、斯拉夫·彼得罗夫（Slav Petrov）、伊利亚·萨茨凯弗（Ilya Sutskever）和杰弗里·辛顿（Geoffrey E.Hinton。语法作为一门外语。在http://arxiv.org/abs/1412.7449, 2014.

[29]奥里奥尔·维尼亚和夸克诉勒。神经会话模型。参加2015年国际机器学习会议：深度学习研讨会。

[30]Sepp Hochreiter和Jurgen Schmidhuber。长期短期记忆。神经计算，9（8）：1735–17801997年11月。

[31]亚历克斯·格雷夫斯、纳夫迪普·贾特利和阿卜杜勒·拉赫曼·穆罕默德。双向LSTM混合语音识别。自动语音识别与理解研讨会，2013年。

[32]萨拉赫·希希和约书亚·本吉奥。长期依赖的递阶递归神经网络。神经信息处理系统，1996。

[33]Jan Koutnik、Klaus Greff、Faustino Gomez和Jurgen Schmidhuber。发条机。2014年国际机器学习会议。

[34]纳夫迪普·贾特利、文森特·万霍克和杰弗里·辛顿。多帧预测的自回归积可以提高混合模型的精度。INTERSPEECH，2014年。

[35]Hasim Sak、Andrew Senior、Kanishka Rao和Francoise Beaufays。快速准确的递归神经网络语音识别声学模型。在INTERSPEECH，2015年。

[36]托马斯·米科洛夫、伊利亚·萨茨凯、陈凯、格雷格·科拉多和杰弗里·迪恩。单词和短语的分布表示及其组成。神经信息处理系统，2013年。

[37]Daniel Povey、Arnab Ghoshal、Gilles Boulianne、Lukas Burget、Ondrej Glembek、Nagendra Goel、Mirko Hannenmann、Petr Motlicek、Yanmin Qian、Petr Schwarz、Jan Silovsky、Georg Stemmer和Karel Vesely。Kaldi语音识别工具包。自动语音识别与理解研讨会，2011。

[38]Jeffrey Dean、Greg S.Corrado、Rajat Monga、Kai Chen、Matthieu Devin、Quoc V.Le、Mark Z.Mao、Marc&apos;Aurelio Ranzato、Andrew Senior、Paul Tucker、Ke Yang和Andrew Y.Ng。大规模分布式深度网络。神经信息处理系统，2012。

# 对齐示例

在本节中，我们将给出我们的模型和注意力分布的其他可视化示例。

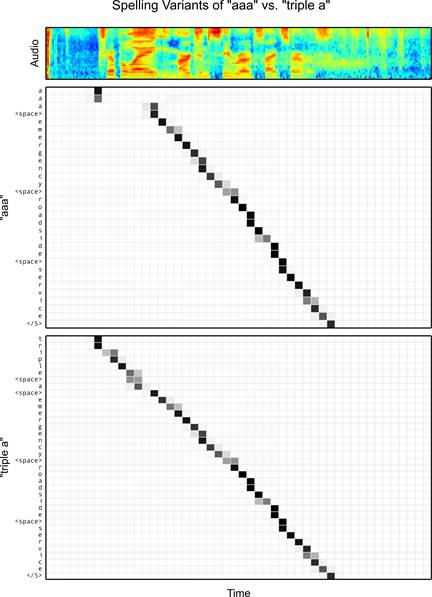


图6：“aaa”和“triple a”的拼写变体产生不同的注意力分布，两种拼写变体都出现在我们的顶梁中。基本事实是：“aaa紧急路边服务”。

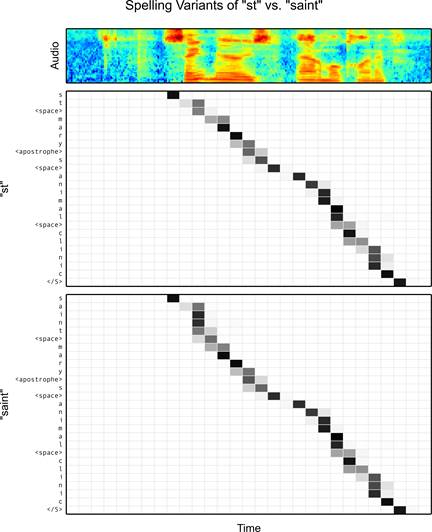


图7：“st”和“saint”的拼写变体产生不同的注意力分布，两种拼写变体都出现在我们的顶梁中。基本事实是：“圣玛丽动物诊所”。

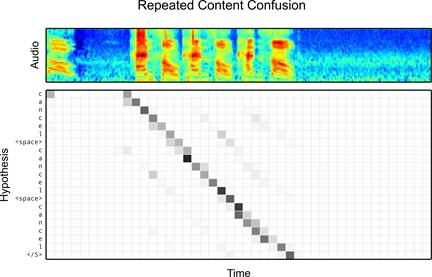


图8：短语“cancel”重复了三次。注意平行对角线，内容注意机制有点混乱，但是模型仍然发出正确的假设。基本事实是：“取消”。