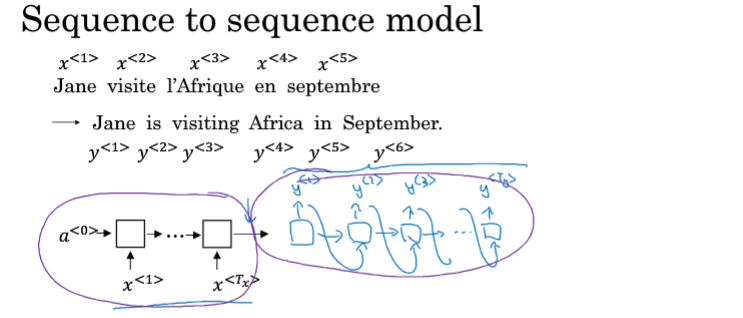
# 第三章 序列模型和注意力机制

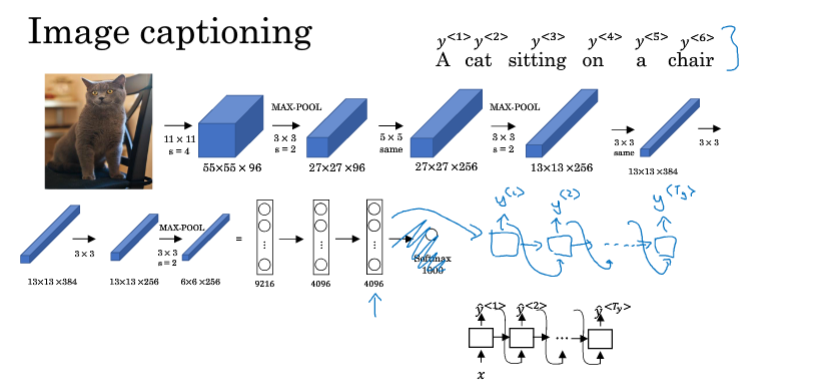
## 3.1 基础模型

这一章要学习sequence to sequence模型，从机器翻译到语音识别，都有很大作用。从最基本的模型开始，再到集束搜索Beam search和注意力模型attention model，一直到最后的音频模型，比如语音。

将一个法语句子翻译成英语，和之前一样，使用到来对应这个句子中的每一个词，使用到来表示输出的英语的每一个单词。如何训练一个网络输入序列x和输出序列y呢，首先建立一个网络，编码网络，encoder network，RNN结构，其单元可以是GRU也可是LSTM，每次只向网络中输入一个法语单词，将输入序列接收完毕后，这个RNN网络会输出一个向量来表示这个输入序列，然后可以建立一个解码网络，它以编码网络的输出作为输入，被训练为每次输出一个翻译后的单词，一直到输出序列的结尾或者句子结尾标记。这个解码网络的工作就结束了。解码网络中每次生成的标记都传递到下一个单元中进行预测，就像之前使用语言模型合成文本一样，深度学习最近的卓越成绩就是这个模型确实有效，在给出足够的法语和英语文本的情况下，训练这个模型，将会很有效。这个模型简单的用一个编码网络来对输入的法语句子进行编码，再用一个解码网络来生成对应的英语翻译。



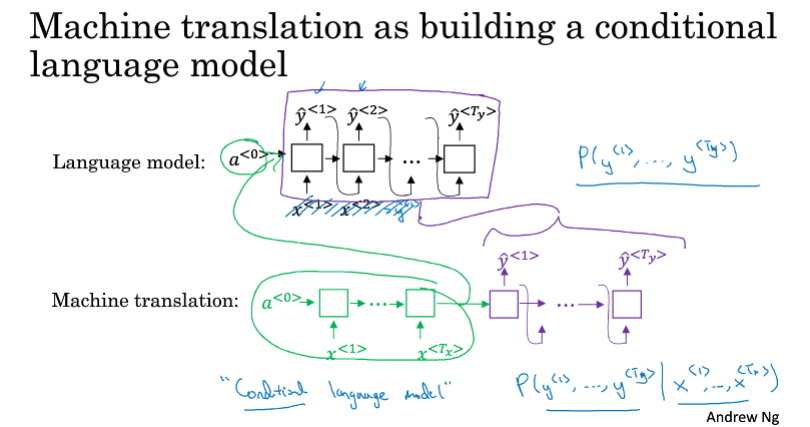
还有一个与此类似的结构被用来做图像描述，给出一张图片，能自动地输出该图片描述的内容，比如，A cat sitting on a chair。如何训练这样的通过输入图像来输出描述的模型呢，方法如下，前面已经学习过如何将图片输入到卷积神经网络中，比如一个预训练的AlexNet结构，然后让其学习图片的编码，或者学习图片的一系列特征，去掉最后的softmax单元，这个结构会输出一个4096维的特征向量，表示的就是这只猫的图片，所以这个去掉softmax层的网络就可以作为图像的编码网络，接着把这个向量输入到RNN中，生成图像的描述，每个时间步生成一个单词。事实证明在图像描述领域这种方法相当有效，特别是当你想要生成的描述不是特别长的时候。



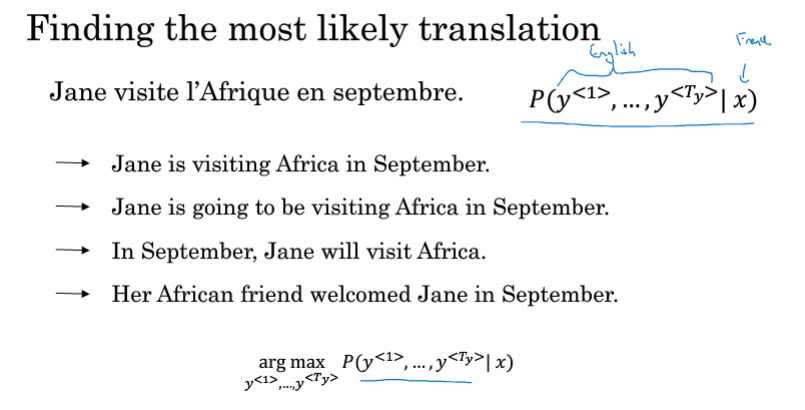
现在知道了基本的sequence2sequence的模型是如何运行的，以及image to sequence模型，不过这两个模型的运作方式有一些不同，主要体现在如何用语言模型合成新的文本并生成对应序列的这方面，一个主要的区别就是大概不会想得到一个随机选取的翻译，想要的是准确的翻译，或者想要最准确的描述。

## 3.2 选择最可能的句子

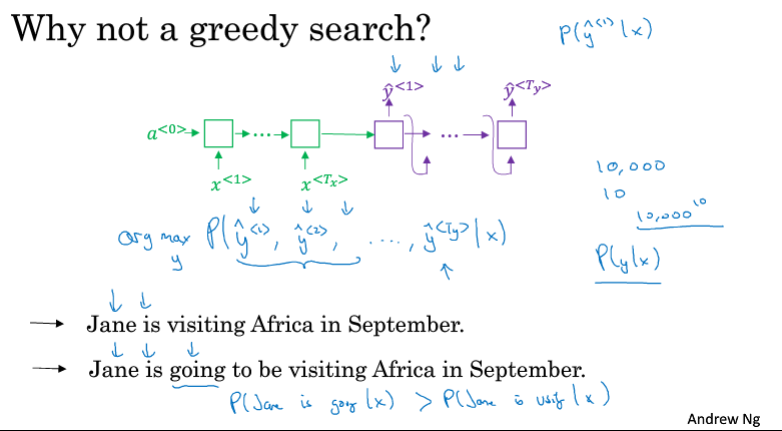
Seq2seq机器语言模型和前面学习的语言模型有很多相似地方，也有许多重要区别。可以把机器语言模型项城市建立一个条件语言模型，在语言模型中，基本RNN模型可以估计句子的可能性，也可以用于生成一个新的句子，有，这个模型中输入,等并不重要，是一个全为0的向量，后面的输入为上一个时间步的输出。而机器翻译模型是，绿色为编码网络，紫色为解码网络，解码网络和基本RNN模型几乎一模一样。两者不同是，语言模型总是以0向量开始，而机器翻译中编码网络会计算出一系列向量来表示输入的句子，并且输入到解码网络中。所以讲机器翻译模型成为条件语言模型。相比语言模型输入任意可能句子的概率，翻译模型会输出句子的英文翻译，这取决于输入的法语句子，,估计一个英语句子的概率，即英语句子相对于法语句子输出的可能性。



翻译模型会告诉你通过模型，不同英语句子相对于输入的法语句子的可能性，x代表法语句子，,显然不想模型随意地进行输出，如果从这个分布中进行采样，得到，可能取样一次，就能得到很好的翻译，但也可能得到一个其他完全不同的翻译，所以当你使用这个模型来进行机器翻译时，并不是从得到的分布中进行随机采样，而是要找到一个英语句子y使得条件概率最大化。所以开发翻译模型时需要想出一个算法用来找出合适的y值，使得该项最大化，，而解决这个问题最常用的就是束搜索（Beam search），了解束搜索之前，可能有疑问，为什么不同贪心搜索呢（Greedy search）。



贪心搜索是一种来自计算机科学的算法，生成第一个词的分布之后，会根据你的条件语言模型挑选出最有可能的第一个词，然后挑选最可能的第二个词然后……但是真正想要的是一次性挑选出整个单词序列，从到，来使得整体概率最大化，所以一个一个地选词的贪心算法并不管用。字典中10000个词，然后加入要输出10个长度的翻译句子，那么可能的句子情况数量非常巨大，不可能计算每一种组合的可能性，所以这时常用的方法就是用一个近似的搜索算法，做的就是尽可能的尽管不一定会成功的将挑选出句子y，使得条件概率最大化。



## 3.3 定向搜索-Beam search

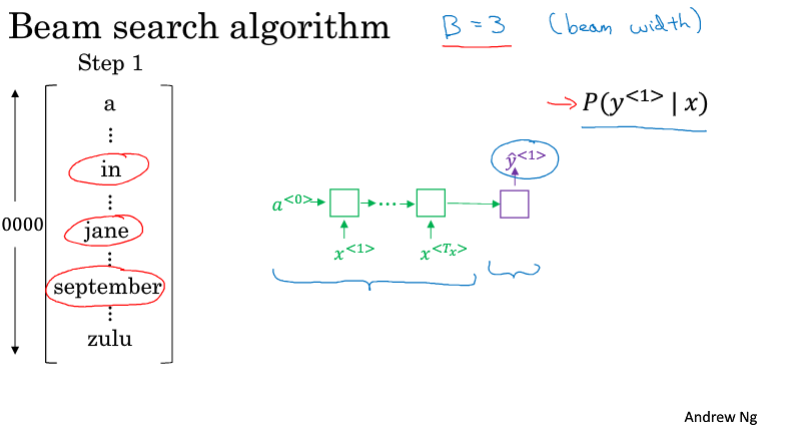
语音识别和机器翻译一样，给定一个输入，不想输出一个任意的句子，而是一个最可能的句子。束搜索就是解决这个的算法，这一节学习如何把集束搜索算法应用到自己的工作中。

使用下面的法语句子为例，翻译为英语，集束搜索首先做的就是挑选出英语翻译中的第一个单词，首先使用seq2seq模型来估计第一个单词的概率值，，贪婪算法只会挑出最可能的那个词然后继续，而集束搜索会考虑多个可能结果，算法有一个参数B，叫做集束宽，设B=3，意味着搜索结果不会只考虑一个可能结果，而是考虑3个，比如对于第一个单词，就会得到前3个最有可能的词，把结果存到内存中，便于后面继续搜索。所以第一步就是通过编码网络将法语句子进行编码，输入到解码网络生成最有有可能的 B个第首位单词。第二步，针对选出来的每个单词考虑第二个词是什么，为了评估第二个词的概率值，就在解码部分，将第一个时间步的输出，并且把in作为第二个时间步的输入，得到，就可以评估第二个词了，得到给定法语句子和第一个词是in的情况下，第二个词的概率，即。注意在第二步更关心的是，要找到最可能的第一个和第二个单词对，所以不仅是第二个词有最大的概率，而是第一个第二个单词组成的单词对有最大的概率，按照条件概率的准则，有下面公式：

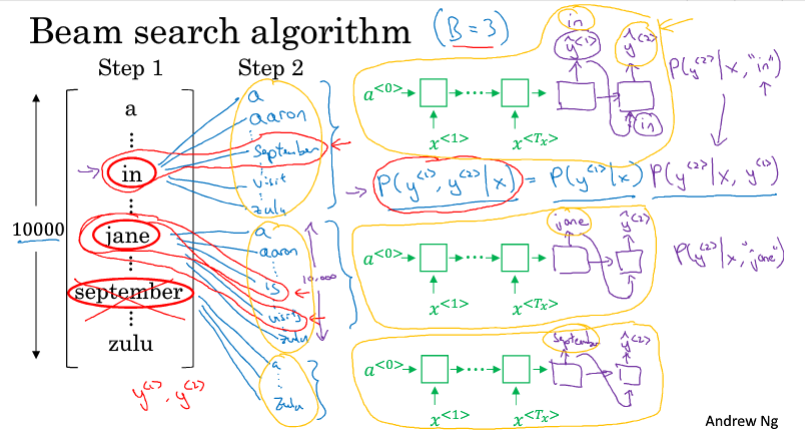
上面是第一个单词是in的情况下如何评估第2个单词的概率，对于剩下的B-1个词也同样的方法查找第二个词。所以第二步时，最终就会有B\*10000中可能的结果了，在这30000种结果中选出前三个，最后又只有3个结果，将30000种可能性减少到集束宽的大小，这三个结果有可能有时相同首位单词的两个。集束搜索会保存这些结果，得到3个（）对，然后用于下一次搜索。因为集束宽为3，每一个时间步的计算都会有3个同样的网络副本来评估部分句子和最后的结果。第三步，此时我们得到了3个的结果，接下里考虑第三个词，还是和第二步一样，有3个网络来计算第三个词，并且第三个时间步的输入为三种可能对的第二个词，提出一个疑问，那如果有第一个词不同，第二个词相同的两个对子，是否第三个词的计算由第三个时间步输入第二个词的结果来计算，后面计算出来内容就会一样呢。不会的，因为第三个词的确定最后不是靠第三个时间步计算出来的分布的简单前3名，而是还要分别乘上前面的对子的概率。所以集束搜索在第三步还是会挑选出，在输入x法语句子，和3个对子的情况下，第三个最可能的词。然后按同样的道理进行第4步……最终这个过程的输出一次增加一个单词，集束搜索就会找到最好的那个句子终止在句尾符号。这里B=3，实际上如果B=1，这就实际上变成了最基本的RNN语言模型，也是贪婪搜索算法。如果使用其他的集束宽值，Beam search就会找到比贪婪搜索更好的输出结果。

简单总结：

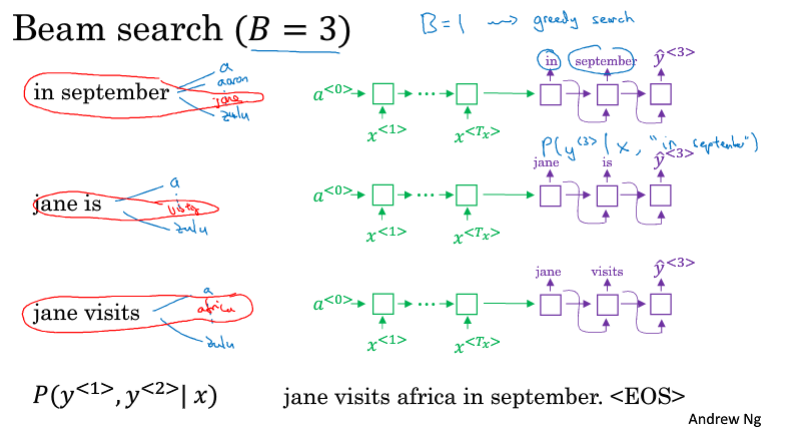
1. 按照普通计算方法，根据softmax结果得到最大的3个结果；



1. 根据3个不同的词，建立三个不同的网络，并且分别计算第二个词的可能结果，最后想要得到3个词对，通过第一个词的概率乘上对应的这个词计算出来的saoftmax概率，就有30000种情况，取其中最大的3种词对。



1. 根据3个词对，同样构建3个网络，分别前两个词依次对应，来计算第3个词，还是有30000种情况，3种词对的概率乘上分别对应的第三个词的softmax计算结果，得到前3名的3个词构成的词组。



1. 依次继续后面的步骤，直到得到结束符。

## 3.4 改进定向搜索

学习了基本的集束搜索算法，再学习一些技巧使算法运行的更好。Length normalization长度化归一可以对集束搜索作调整，得到更好的结果。关于集束搜索，总目标就是要最大化这个概率，其实就是输入x情况下，最好的词组合起来的概率，这个符号表示连乘。

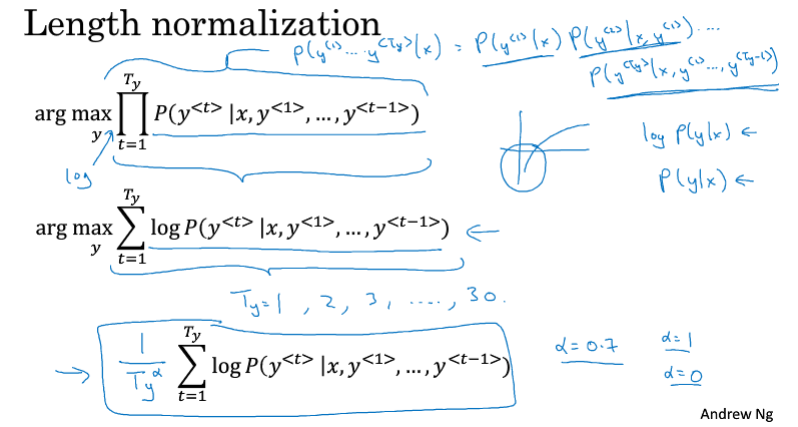
（1）

这个公式展开是最后一步的计算公式，前多少个项乘就是第一个时间步确定第几个词的公式。每一项都是小于1的值，或者说远小于1，所以这么多的数乘起来可能会造成数值下溢，就是数值太小导致电脑的浮点表示不能精确地储存，所以实际中不会真的最大化这个公式，而是取log值，那么乘积就变成了求和，所以目标转化成最大化这个log求和的概率值：  
 （2）

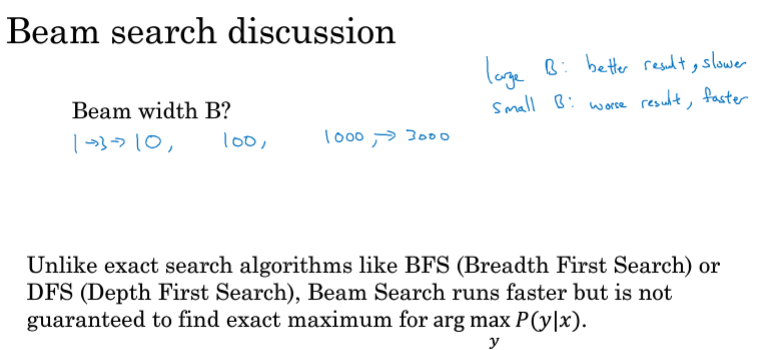
这两个公式在选择最可能的y句子时，会得到一样的结果，通过取log会得到一个数值上更加稳定的算法，不容易出现四舍五入的误差。对于新的公式，还可以做一些改变，使得机器翻译表现的更好。对于原来的目标函数，如果目标句子很长，那么概率的乘积下来就会很小，所以这个函数的缺点就是可能不自然地倾向于简短的翻译结果，因为由少数单词组成的短句子的概率的结果不会那么小。并且新的公式同样有这样的问题，log结果都是负值，越加值越小。所以改变就是不再最大化（2）这个目标函数，将其归一化，通过除以翻译结果的单词数量，取每个单词概率对数值的平均，明显减少对输出长的单词的影响。实践中，相比除以用一个更柔和的方法，在上加上指数，如果，就相当于完全用长度归一化。

（3）

如果，就相当于没有归一化。就是算法的另一个参数，需要调整大小来得到最好的结果，这样使用没有理论验证，但效果确实很好。尝试不同值，看哪个效果更好。



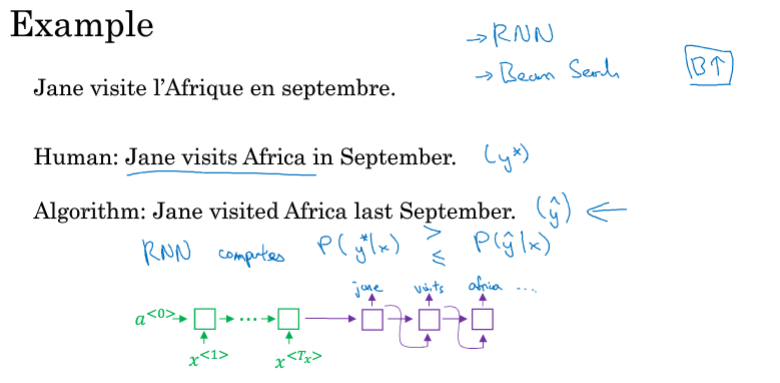
总结下如何运行束搜索算法，运行的时候会看到很多长度不同的句子，加入B=3，，那么就会得到1,2，，，，30长度的三个可能的句子，针对这些可能的输出句子，用新的公式（3）给它们打分，取概率最大的几个句子，然后对这些句子计算（3）这个目标函数，最后从经过评估的句子中挑选出在归一化的log概率目标函数上得分最高的一个。这也叫做，归一化的对数似然目标函数。



实现细节，如果选择B，B越大，考虑的结果越多，找到的句子可能越好，但是算法的计算代价越大，计算慢，内存占用多，要把很多的可能结果保存起来。前面例子中B=3，实践中这个值偏小，一般B=10，这也取决于不同应用。在自己的应用中，应该尝试不同的B值，随着B增大，性能提高会越来越小。束搜索算法不想其他计算机科学里的算法，如深度优先搜索和广度优先搜索等，这些都是精确的搜索算法，束搜索运行的更快，但是不能保证一定找到argmax准确的最大值。在束搜索上做误差分析是很有用的工具之一，可以通过简单的计算来指导如何改进算法，是否应该增大B，或者应该设置为多少。

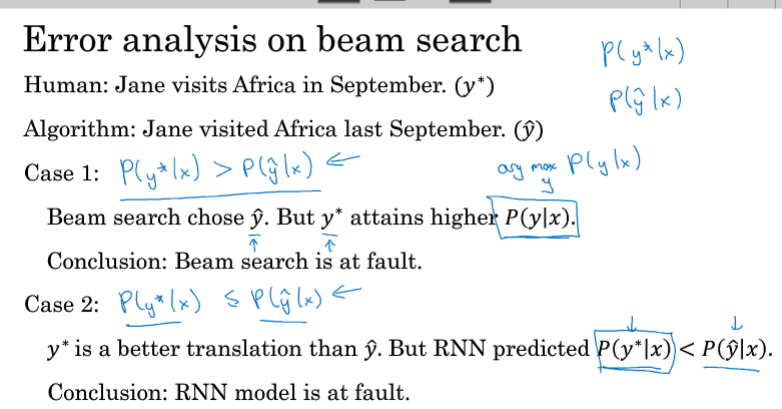
## 3.5 定向搜索的误差分析

第三门课中讲过，误差分析如何帮助集中时间做项目中最有用的工作。束搜索是一种近似搜索算法，也称作启发式搜索算法，不总是输出最好的结果，而是保存着几个可能最好的结果，束搜索出现错误是什么样的呢？

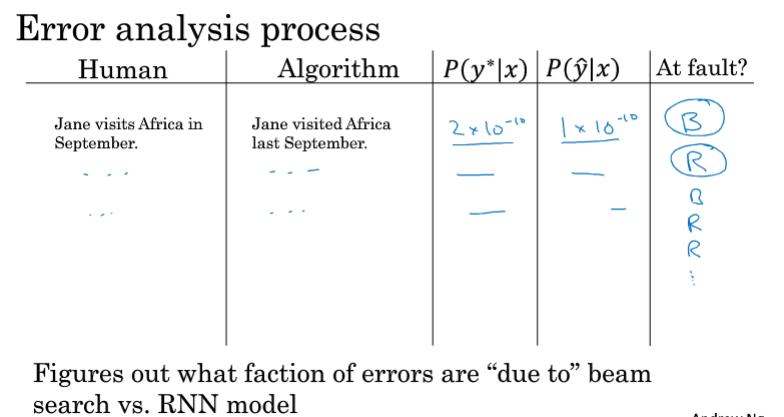


误差分析和束搜索算法是如何相互作用的，如何确定是束搜索出现问题需要修正还是RNN模型出了问题。对束搜索算法进行误差分析：对于之前那个法语例子，假设在机器翻译的dev开发集中，人工是这么翻译的：Jane visits Africa in September 将这个标记为。假设语言模型中翻译的结果是：Jane visits Africa last September这是个完全改变意思的错误翻译。你的模型有两个部分，RNN网络生成句子和Bean搜索部分，以某个宽度B运行，可能需要找出哪部分出的错。大家可能想着搜索更多的训练数据，总归没什么坏处，或者增大B，也是不会错的。但是单纯的收集更多的训练数据或者增大B可能不会得到预期的表现结果，但是如何知道是否应该花时间改进搜索算法呢？

分解问题，RNN会计算，此时最有效的事是用模型计算的同时计算，比较哪个值可能更大。如果，说明束搜索选择了，也说明束搜索不能给一个能使最大化的y值，所以是束搜索出错了。如果，是比更好的翻译结果，可是值却小，说明是RNN出了问题。少讲了一些关于长度化归一的细节，如果使用了长度化归一，可能不是比较和的值，而是比较长度化归一后的最优化目标函数值，不过现在忽略这种复杂的情况。



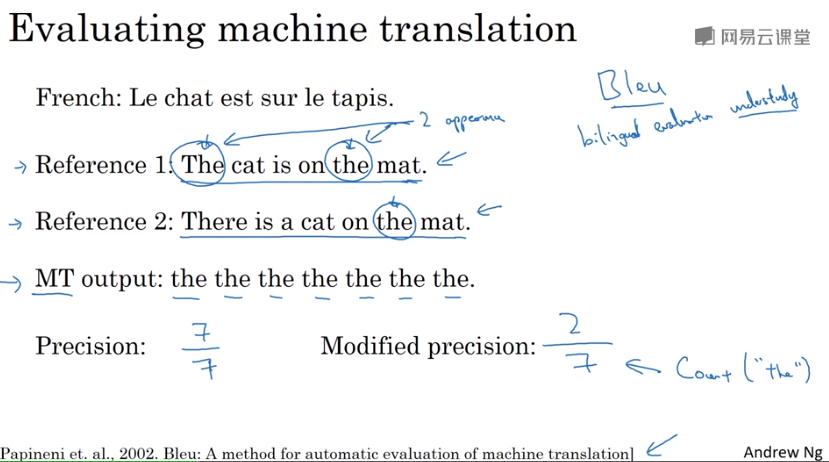
虽然误差分析过程看起来就像下图这样，先遍历训练集，然后找出算法产生的错误，接着继续遍历第二个错误，或者说第二个由算法输出的糟糕的结果，再来看这些可能性，再遍历其他问题。通过这个过程，就饿能够通过执行误差分析，得出束搜索算法和RNN模型出错的比例是多少。就可以对开发集中所有模型输出的句子比人工翻译差的句子的错误，尝试确定这些错误，是哪存在问题。哪个更可能是出错的原因，花时间解决，如果是束搜索算法造成大部分错误，增大集束宽度，如果是RNN模型，进行更深层次的分析，来决定是否需要增加正则化还是获取更多的训练数据，又或是尝试一个不同的网络结构或其他方案。



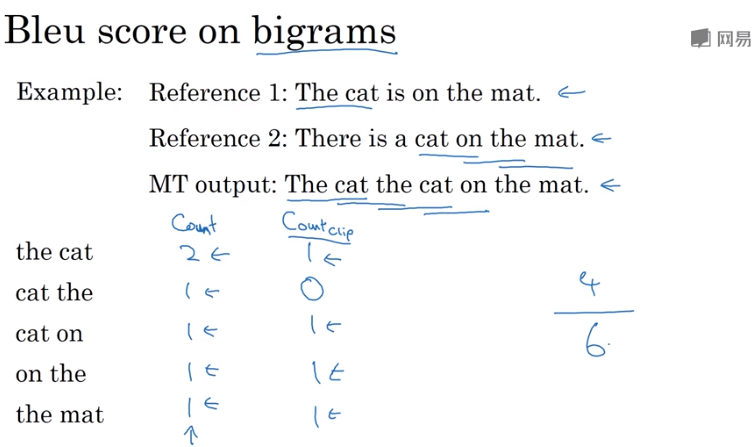
## 3.6 Bleu得分（选修）

机器翻译最大的挑战是对应一个法语句子，可以有都比较好的多种翻译，有多个同样好的答案时，如何评估一个机器翻译系统呢。不像图像识别只要测量准确性，如果有多个答案，如何衡量准确性呢，常见的解决解决办法就是Bleu得分。

Bleu, bilingual evaluation understudy双语评估代替，意思是可以代替人工评估者来评估机器翻译结果的每一个结果。Bleu得分所做的是给定一个机器生成的翻译，能够自动的计算一个分数来衡量机器翻译的好坏。直观上，只要机器翻译结果与任何一个人工翻译的结果足够接近，就会得到一个高的Bleu分数。其理念是观察机器生成的翻译，然后看生成的词是否在至少一个人工翻译参考之中，因此人工翻译结果会包含在开发集或是测试集中。

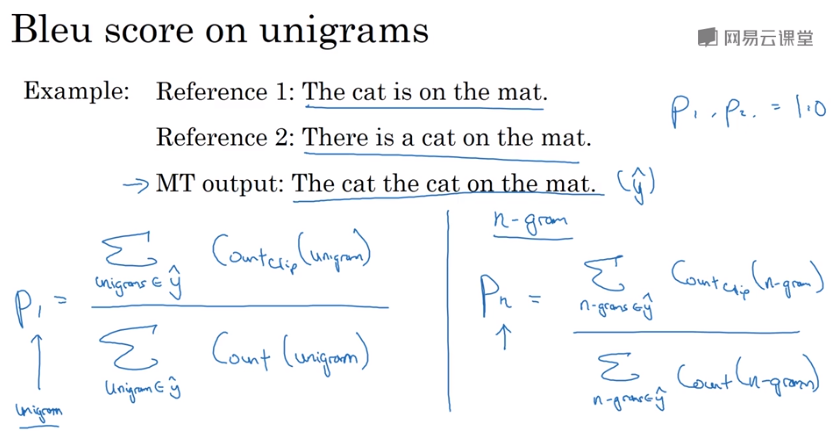


假设对于图中的案例一个比较糟糕的输出结果为：the the the the the the the，衡量机器翻译输出质量的方法之一是，观察输出结果的每一个词看其是否出现在参考翻译中，即为机器翻译的精确度，这种情况下，结果7个the都出现在了参考中，看上去每个单词都很合理，精确度为，很明显这样的结果并不是很有用，所以使用改良后的精确度评估方法，把每一个单词的计分上限定位它在参考句子中出现的最多次数，也就是说，the最多在某个参考句子中最多出现两次，所以最多只能记两分，你那么精确度就为。改良后精确度公式为：



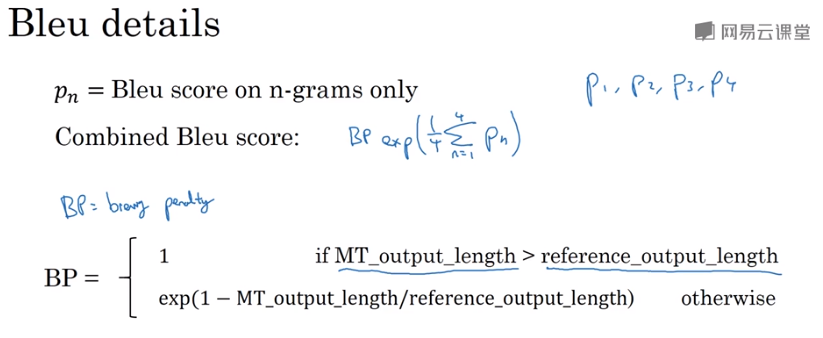
到目前为止，只考虑了单个单词，在Bleu得分中，如果想考虑成对的单词，接下来定义二元词组Bleu得分，并且这只是最终Bleu得分的一部分，最终得分会考虑一元词组，二元词组，同时也许会有更长的单词序列。二元词组，即相邻的两个词，对于精确度计算，分母为所有可能的相邻词组的出现总次数，分母为每个词对应的在参考中出现的计分之和。表示n元词组的精确度，n-gram代表n元词组。

总公式：



这些改良后的精确度的评估对象是一元词组或二元词组或者n更大的n元词组，这个方法都能够衡量机器翻译中与参考相似重复的程度，另外，如果机器翻译与参考1或2完全一致的话，所有P的值都会为1。另外，结果为1不定义完全与参考相同，可能是另外的组合形式，但是翻译效果一样好。

最后将这些结合构成最终的Bleu得分，首先对所用的P取平均值，再对这个结果进行e为底的指数运算，exp()表示e为底的指数运算。所以假设考虑到4元，那么就有，，，，严格强调递增的运算。实际上还会用一个BP成大因子来调整，brevity penalty 简短惩罚，如果你输出一个非常短的翻译，会更容易得到一个更高的精确度，因为输出的大部分词可能都出现在参考之中，但我们并不想要偏向更短的结果，所以需要一个调整因子。

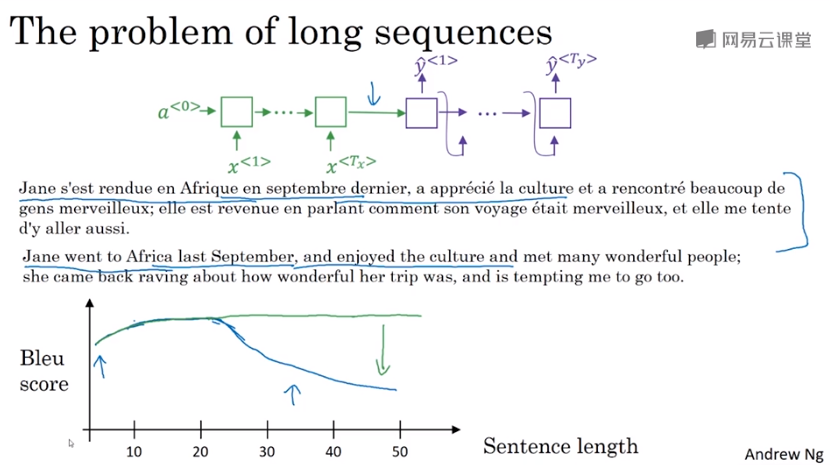


在以前的学习中，了解了有单一实数评估指标的重要性，能够让你尝试两种想法，然后看哪一个得分更高，尽量选择得分更高的那个。Bleu得分对机器翻译有重要作用就是因为是一个不是很完美但是很好的单一实数指标。一般不会自己实现，直接下载开源代码，用于自己的机器翻译系统评估。Bleu被广泛用于生成文本的评估，图像描述系统也会用到。语音识别中通常不用，因为语音识别中通常只有一个答案，可以用其他的评估方法，来看你的语音识别结果是否十分相近或者字字正确。

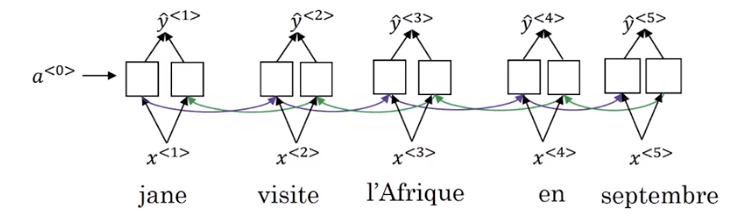
## 3.7 注意力模型直观理解

对于之前的编码解码机器翻译模型，要对其做一些改变，称为注意力模型，Attention model，这会使其工作的更好。注意力模型或者说注意力这种思想，已经是机器学习中重要思想之一。

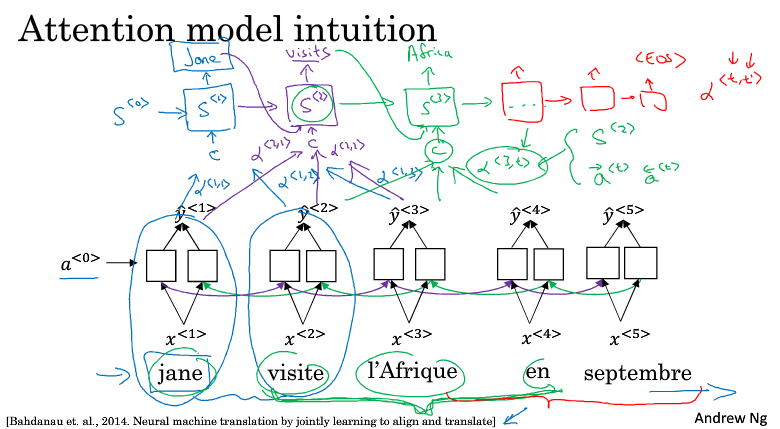
人工翻译并不会通过读整个法语句子再记忆里面的东西，然后从零开始，机械式地翻译成一个英语句子，而人工翻译首先做的可能是先翻译出句子的一部分，在看下一部分，一直这样下去，会通过句子，一点点翻译，因为记忆整个的长的句子是非常困难的。在编码解码结构中，会看到它对于短句子效果非常好，有一个相对高的Bleu分，但是对于长句子，表现就会变差。随着单词数量的变化，短的句子会难以翻译，因为很难得到所有词。对于长的句子效果也不好，因为在神经网络中记忆长的句子是非常困难的。接下来会见识到注意力模型，翻译得很像人类，一次翻译句子的一部分。而且有了注意力模型，机器翻译的表现就会像绿色曲线一样，因为翻译只会翻译句子的一部分，就不会出现衡量神经网络记忆长句子能力的下跌。



关于注意力机制运行的一些直观东西，虽然这么模型源于机器翻译，也可以推广到其他领域，用一个短句来举例说明，实际上遇到的句子更长一些。对于下图中一个比较短的法语句子，假设使用一个双向的RNN，来计算每个输入单词的特征集，但我们要做的不是只翻译一个单词，所以先去掉RNN模型上面的Y输出。

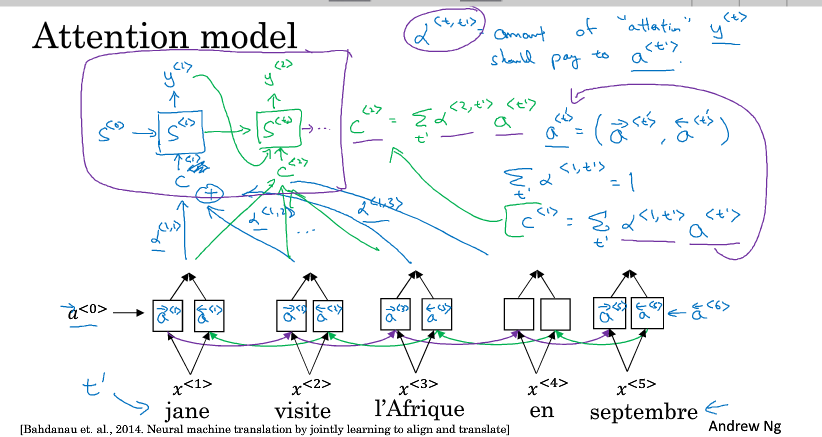


要对单词做的是对于句子里面的每五个单词，计算一个句子中单词的特征集，也有可能是每个词周围的词，接下来开始生成英文翻译，将使用到另一个RNN，对于上面的RNN不用A来表示隐藏值感知机，用S来表示隐藏状态。希望在上面这个模型生成的第一个单词将会是Jane。所以当你在尝试第一个词输出时，应该看输入法语句子的哪个部分，或许你应该看第一个词或它附近的词，但不需要看的太远，比如句尾，所以注意力模型就会计算注意力权重，用来表示生成第一个词时应该放多少注意力在第一块信息处，表示计算第一个词时放多少注意力在第二个法语词信息上，这里的就叫做注意力权重，说明了应该花多少注意力在记号为C的内容上，这是RNN的第一个单元如何尝试生成第一个词。对于要生成的第二个隐藏单元，用新的记忆权重提示什么时候生成第二个词，后面的单词同理，不会看到句子的最后一个词。输出的第一个词Jane也会输入到第二个隐藏状态，以及需要花注意力的上下文，一起生成第二个词。关于细节，如何准确的定义上下文，即考虑单词前后多少范围内的单词，以及用到的公式，和如何计算注意力权重。取决于在t时的双向RNN的激活值以及后面的词的激活值，和上一步隐藏状态S值，由这些一起影响应该花多少注意力在输入的法语句子的某个词上面。RNN每次前进生成一个词，直到最终生成。



## 3.8 注意力模型

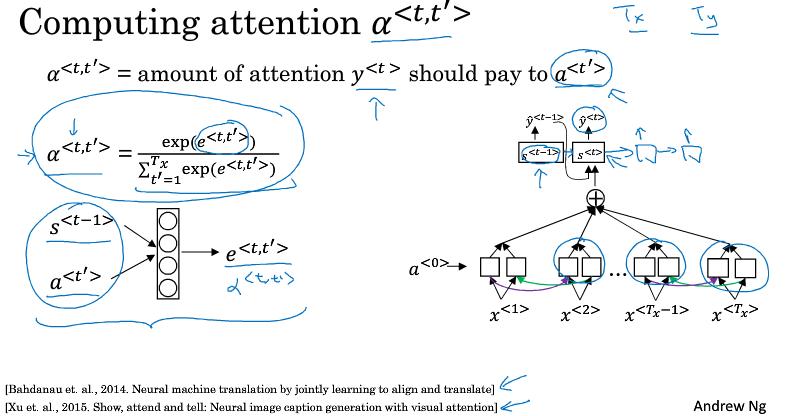
上一个视频讲了注意力模型如何让神经网络只注意到一部分的句子输入，当它在生成句子的时候，更像人类翻译，将这些想法转化为式子来应用注意力模型。先假定有一个输入句子，并使用双向的RNN或者双向的GRU或者双向的LSTM去计算每个词的特征，实际上GRU和LSTM经常应用于这个。



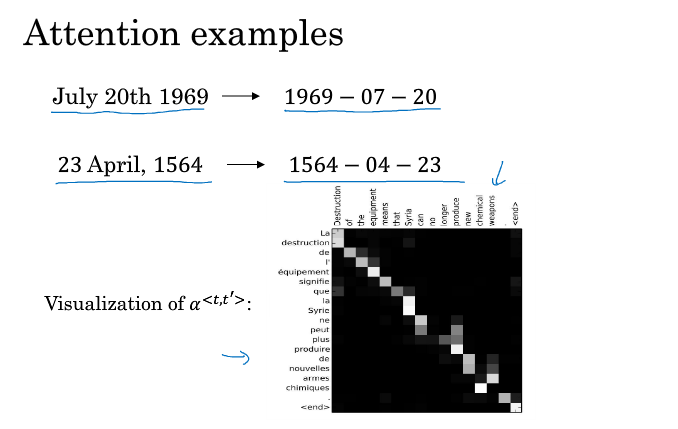
对于前向传播，有第一个时间步的前向传播的激活值，第一个时间步后向传播的激活值。第t个时间步的激活值用来一起表示这些联系，就是时间步t上的特征向量，用状态S表示生成翻译，注意力参数会告诉我们这个生成的上下文需要取决于多少我们得到的特征，或者不同时间步上计算出来的激活值。对于：

用在双向RNN上每个输入的激活值值进行索引，表示生成第一个翻译的词时，在个输入词上的记忆权重，为双向RNN上第步的激活值。关于注意力权重，在需要考虑的值上的权重值之和为1。这是生成的第一步，然后进入下一个时间步，得到第二个输出。上半部分的网络相当于标准的RNN序列，有上下文向量作为输入，一次一个词的生成翻译，也定义了如何通过这些注意力权重和输入句子的特征值来计算上下文向量。剩下要做的就是定义如何计算这些注意力权重，公式：

首先要计算，使用softmax函数来确保这些权重加起来等于1。如何计算呢，根据我们思考，假设把网络简化成一个小的神经网络，在计算的时候，通常要用到上一个单元的隐藏值，以及各个输入的激活值，所以需要去学习这个函数到底是什么，反向传播和梯度下降会学习到一个很好的函数，这表示如果你应用这整个的模型，然后用梯度下降来训练它，这是可行的，这个简单的神经网络就告诉你应该花多少注意力在上面。

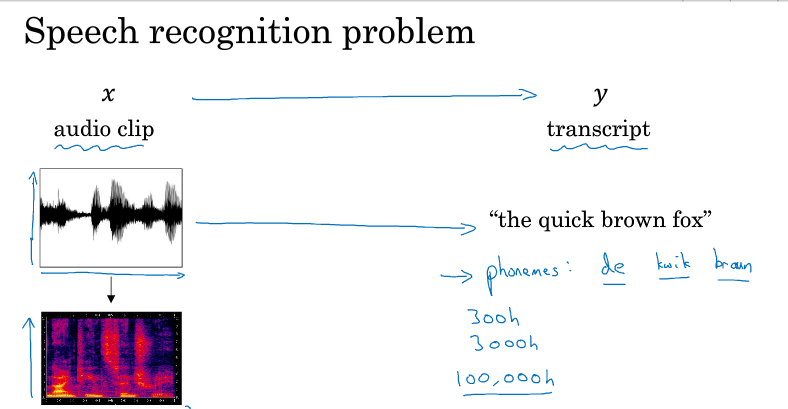


这个模型的缺点就是复杂度比较高，，如果有个输入的单词和个输出单词，于是注意力参数的总数就是。这个想法也用到了给图片写标题的应用，在写标题的时候，一次只看一部分图片。注意力模型的例子还有标准化日期格式，训练一个神经网络，输入任何形式的日期，使用注意力模型生成标准化的日期形式。

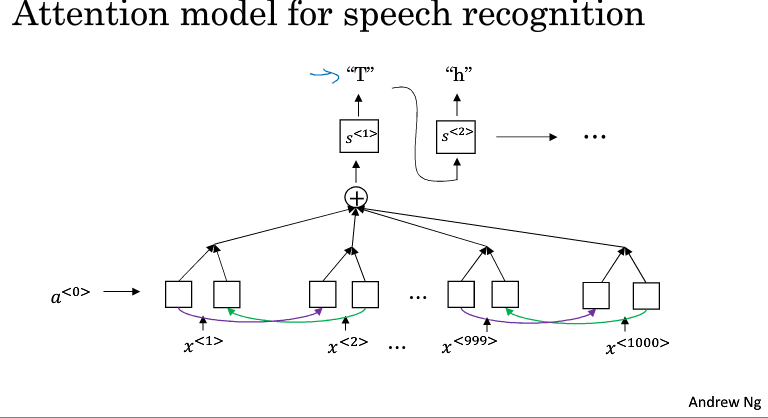


## 3.9 语音辨识

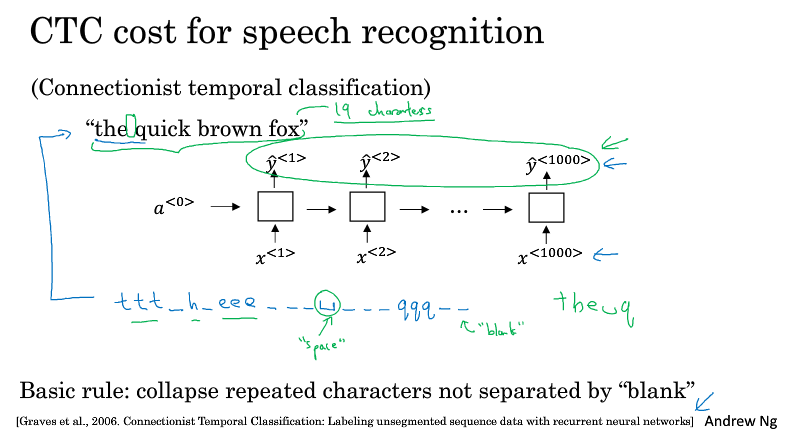
Seq2seq在语音识别的准确性上有了很大的提升，如何应用于音频数据呢。有一个音频片段x，任务是自动生成文本y，音频图横轴是时间，竖直上是气压变化，加入这个音频片段的内容是“the quick brown fox”，考虑到人的耳朵并不会处理声音的原始波形，而是通过一种特殊的物理结构来测量不同频率和强度的声波，音频数据的常见预处理步骤就是运行这个原始的音频片段，生成一个声谱图，横轴是时间，纵轴是声音的频率，图中不同的颜色显示了声波的大小，也就是在不同的时间和频率上这些声音有多大。通过这样的声谱图或者伪空白输出，也经常应用于预处理步骤，音频被输入到学习算法之前，而人耳所做的计算和这个预处理过程非常相似。曾经有一段时间，音频是用音位来构建的，人工设计的基本单元，语音学家认为用这些基本音位来表示音频是做语音识别最好的办法，不过在end-to-end模型中，发现这种音位表示法已经没必要的了。而是通过一个系统，直接输入音频得到相应的文本。



用一个很大的数据集，语音识别的数据集可能长达300个小时，甚至3000小时都是合理的，大量数据集中可能由几千种不同的声音，而且最好的商业系统已经训练了超过1万个小时的数据。在文本音频数据集中同时包含x和y，通过深度学习算法大大推进了语音识别的进程。



提到过注意力模型，在横轴上，也就是输入音频的不同时间帧上，可以用一个注意力模型来输出文本描述。另一个就是用CTC损失函数来做语音识别，Connectionist Temporal Classification，算法思路是假设语音片段内容是某人说“the quick brown fox”，使用一个新的网络结构如图，这里画的比较简单，实际可能是双向的LSTM等并且通常是很深的模型，注意一下这里时间步的数量非常的大，在语音识别中，通常输入的时间步数量要比输出时间步的数量多出很多，比如说假如有一段10秒的音频，频率为100Hz，那么就有1000个输入，那么怎么办。CTC允许RNN生成这样的输出，如下，这样的输出也被看成是正确的输出，CTC损失函数的一个基本规则是将空白符之间重复地字符折叠起来，这样一来神经网络就可以有1000个输出了，因为有很多这种重复地字符和很多插在其中的空白符，所以最后得到的文本会短很多。通过这种方法，使网络强制输出1000个输出来表示正确的这段19个字符长的输出。



## 3.10 触发字监测

随着语音识别的发展，很多设备都可以被关键字唤醒，这个就是触发字检测系统，对于使用什么算法最好还没有定论。现在有这么一个RNN，要做的就是把一个音频片段计算出它的声谱图特征，得到特征向量x1，x2，x3等输入到RNN中，最后要做的就是定义目标标签y，就可以在训练集中把目标标签都设置为0，在关键字的那个点把目标标签设为1，在这之后某个点，关键字又被说了一次，同样也把这个点设置为1，这样的标签方案对于RNN来说是可行的。不过这个算法有一个明显的缺点，构建了一个很不平衡的训练集，0比1多了很多，还有一个解决方法训练起来还是很有效的，比起只在一个时间步上输出1，其实可以在输出变回0之前多次输出1，比如在固定的一段时间内输出多个1，这样的话就提高了输出1和0的比例。比如在一个音频片段上，在说出关键字后的一段，让RNN多次输出1。

