中国科学院自动化研究所
INSTITUTE OF AUTOMATION
CHINESE ACADEMY OF SCIENCES

语音交互技术
——语音识别（三）

中国科学院自动化研究所
模式识别国家重点实验室

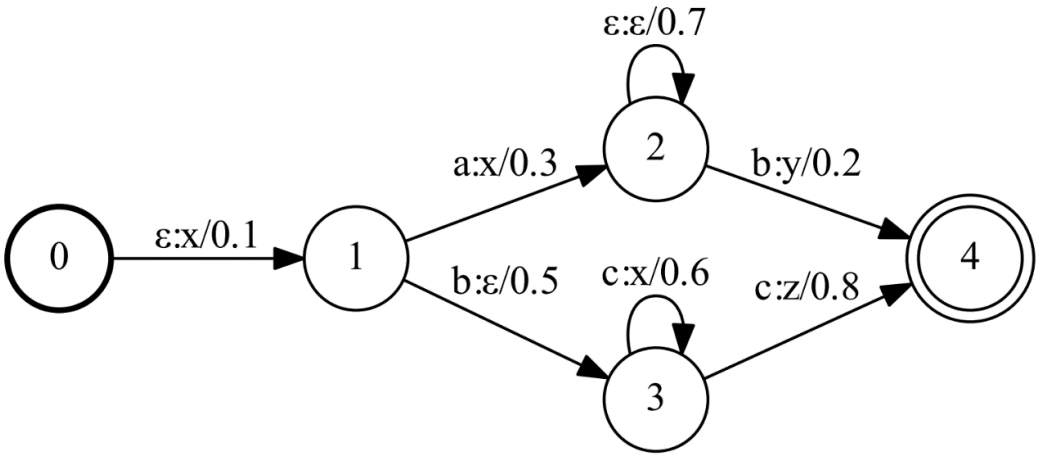
陶建华
jhtao@nlpr.ia.ac.cn

提纲

- 解码空间构建
  - WFST回顾
  - 解码图的构建
- 解码搜索
  - 维特比搜索
  - 束搜索
  - 解码实例
- 端到端语音识别模型
  - 为什么要进行端到端语音识别
  - 循环神经网络转换器
  - 基于注意力机制的端到端

回顾加权有限状态转换器

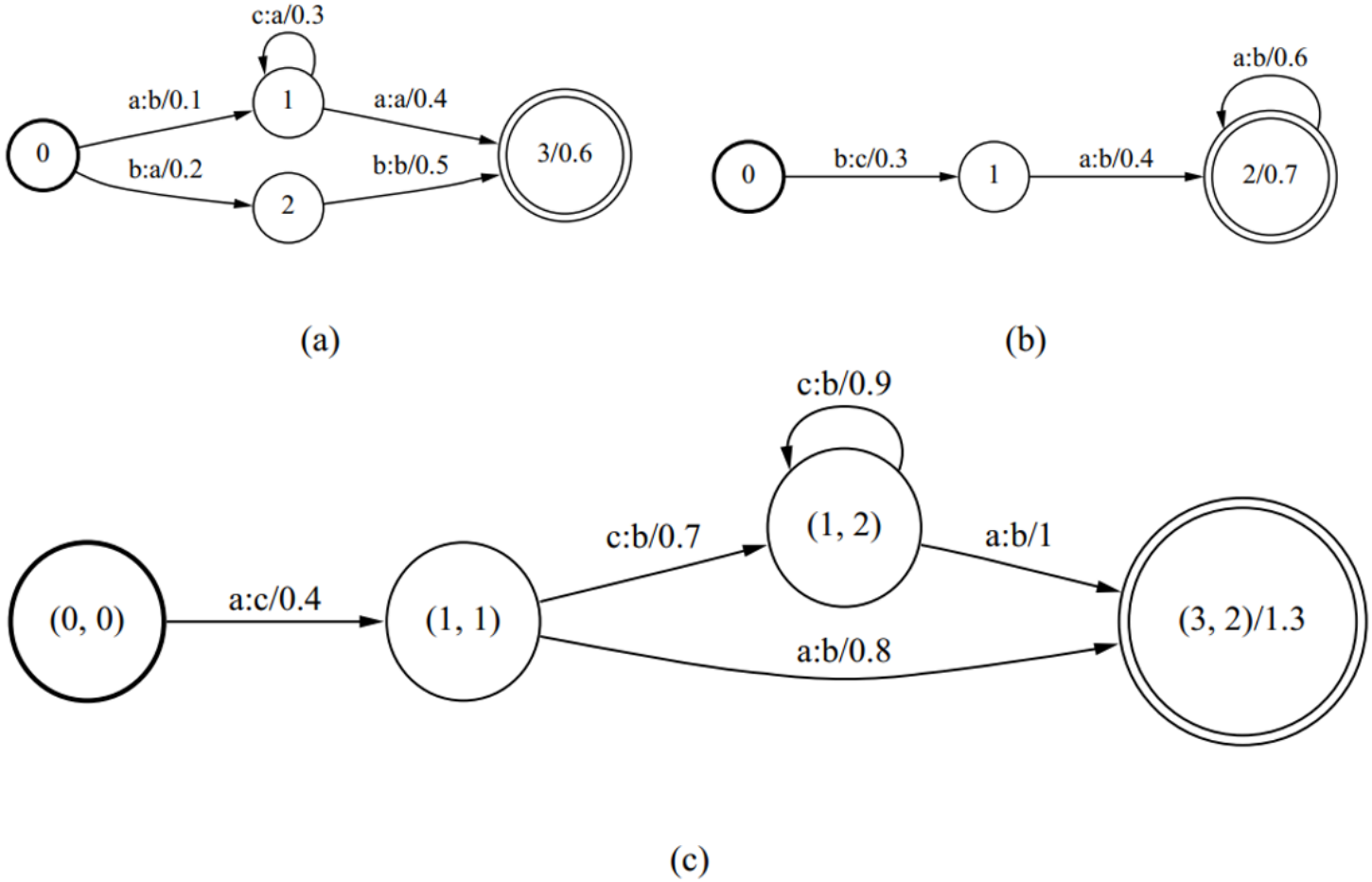
- 加权有限状态转换器是一种图(Graph)。



- 图的节点(node)表示状态(state)，边表示输入符号，输出符号，以及这条边的权重。
- 可以表示序列到序列的转换。

回顾加权有限状态转换器

- 合并操作：组合不同层级的知识。



回顾加权有限状态转换器

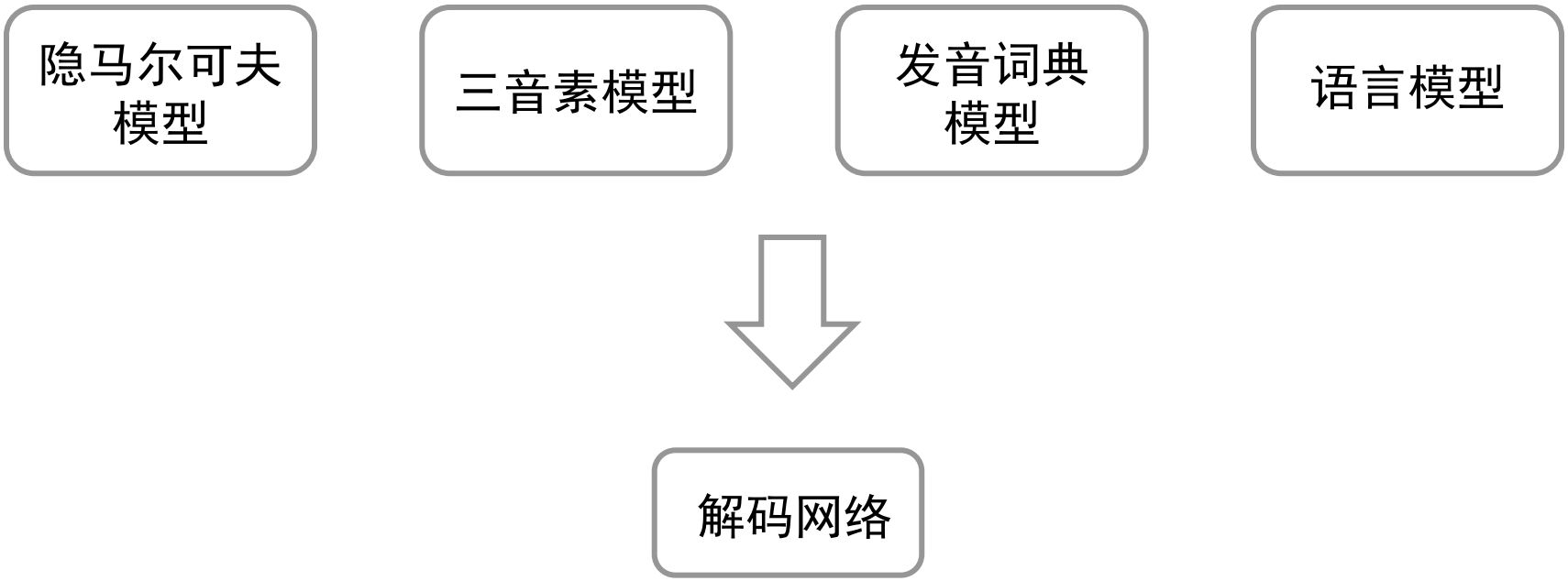
- 如果一个加权有限状态转换器只有一个初始状态，并且同一状态的任意两个转移，输入符号都不同，那么这个转换器是确定的。
- 确定化就是将一个转换器转换为等价的确定的转换器。
- 最小化操作就是将一个转换器转换为等价的转换器中状态数和转移数最小者。

提纲

- 解码空间构建
  - WFST回顾
  - 解码图的构建
- 解码搜索
  - 维特比搜索
  - 束搜索
  - 解码实例
- 端到端语音识别模型
  - 为什么要进行端到端语音识别
  - 循环神经网络转换器
  - 基于注意力机制的端到端

解码空间构建

- 基于加权有限状态转换器融合不同层级的知识。



语言模型的自动机表示

- 给定一个字符序列 $w_1,...,w_T$ ，则根据一个N元语法语言模型估计其发生的概率为

$$P(w_1,...,w_T) = P(w_1)P(w_2|w_1)...P(w_N|w_1,...,w_{N-1}) \prod_{t=N+1}^{T} P(w_t|w_{t-N+1},...,w_{t-1})$$

- 语言模型的自动机表示，就是希望构建一个加权有限状态自动机，当一个字符序列输入到自动机中并被接收，路径上的总权重可以表示此句子发生的概率。

语言模型的自动机表示

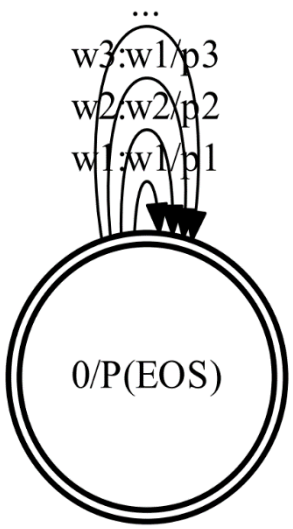
- 考虑一元语法

$$P(w^i) = p_i,\ i = 1, \ldots, L$$

- 一个字符序列的在一元语法下发生的概率可以表示为

$$P(w_{i_1}, \ldots, w_{i_T}) = \prod P(w_{i_t})$$

- 可以构建这样的WFST

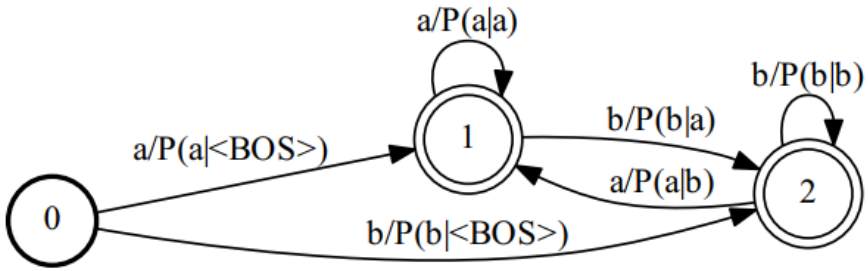


语言模型的自动机表示

- 考虑一个不含回退的2元语法语言模型，且给定词表中任意一个词，另一个词的概率都不为0。
- 考虑一个简单的情形：词汇表中只有a和b两个词，于是语言模型中存在如下概率

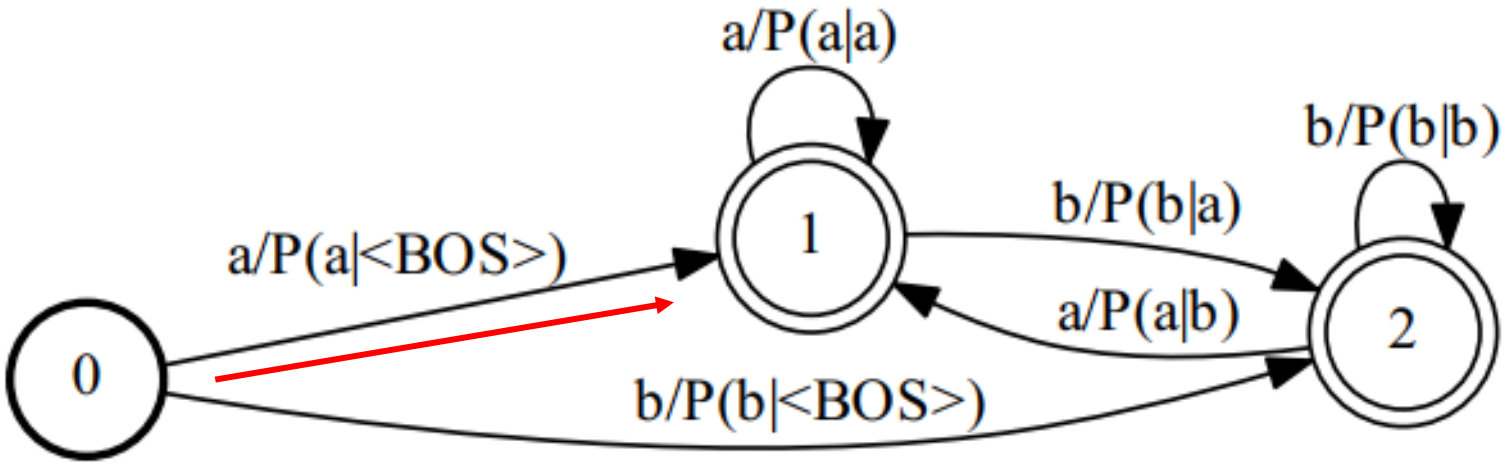
$P(a\|<BOS>)$	$P(a\|a)$	$P(a\|b)$	$P(<EOS>\|a)$
$P(b\|<BOS>)$	$P(b\|a)$	$P(b\|b)$	$P(<EOS>\|b)$

- 于是可以构建这样的自动机



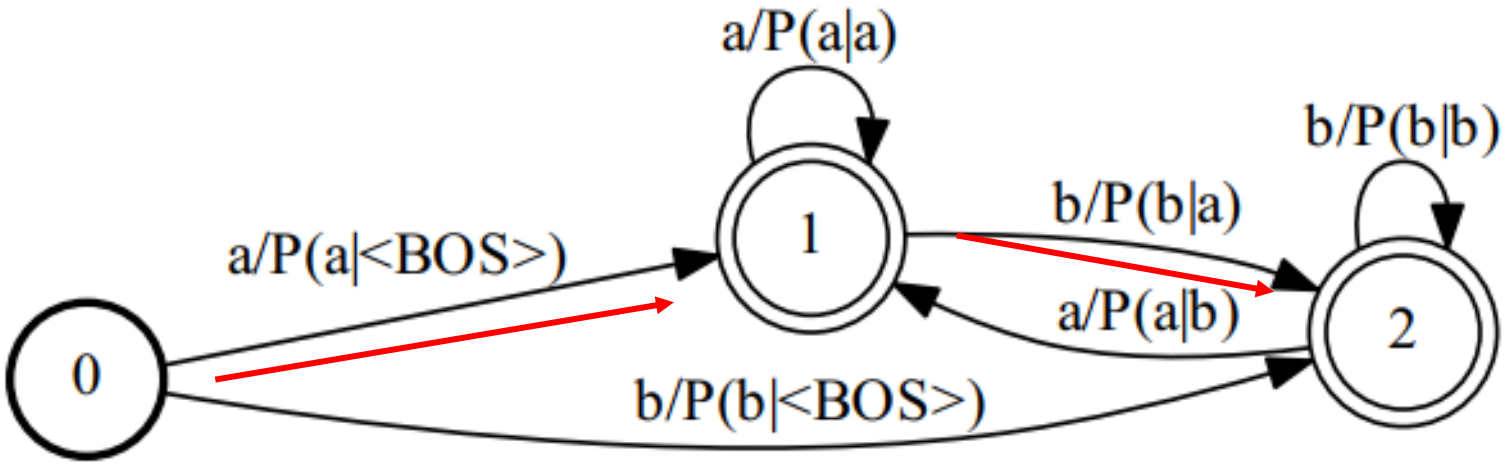
语言模型的自动机表示

- 例子：考虑句子a b a a b



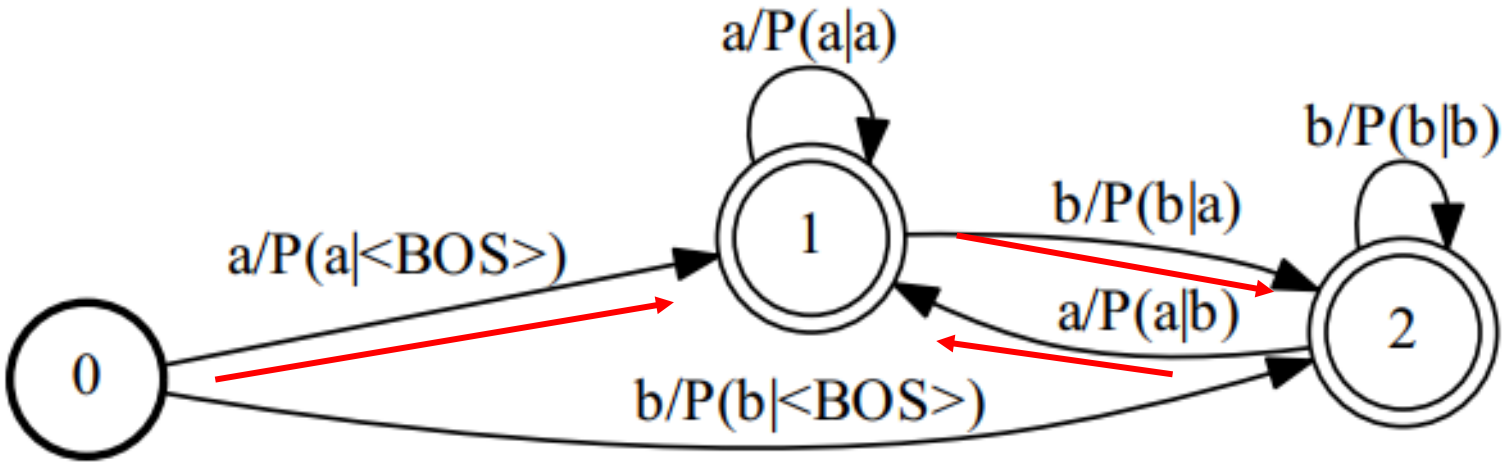
语言模型的自动机表示

- 例子：考虑句子a b a a b



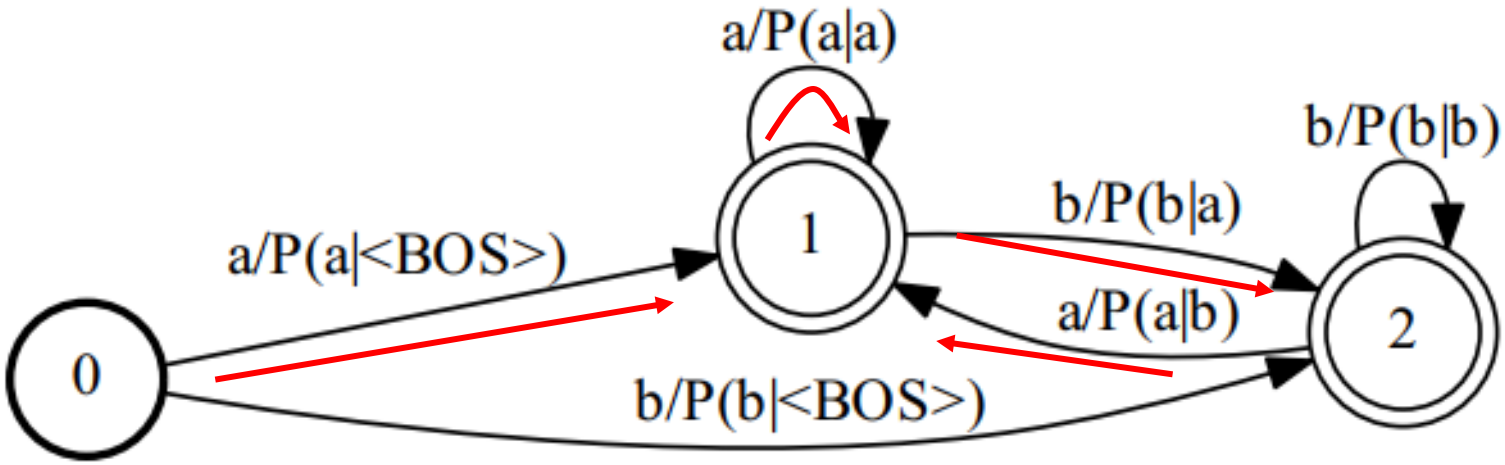
语言模型的自动机表示

- 例子：考虑句子a b a a b



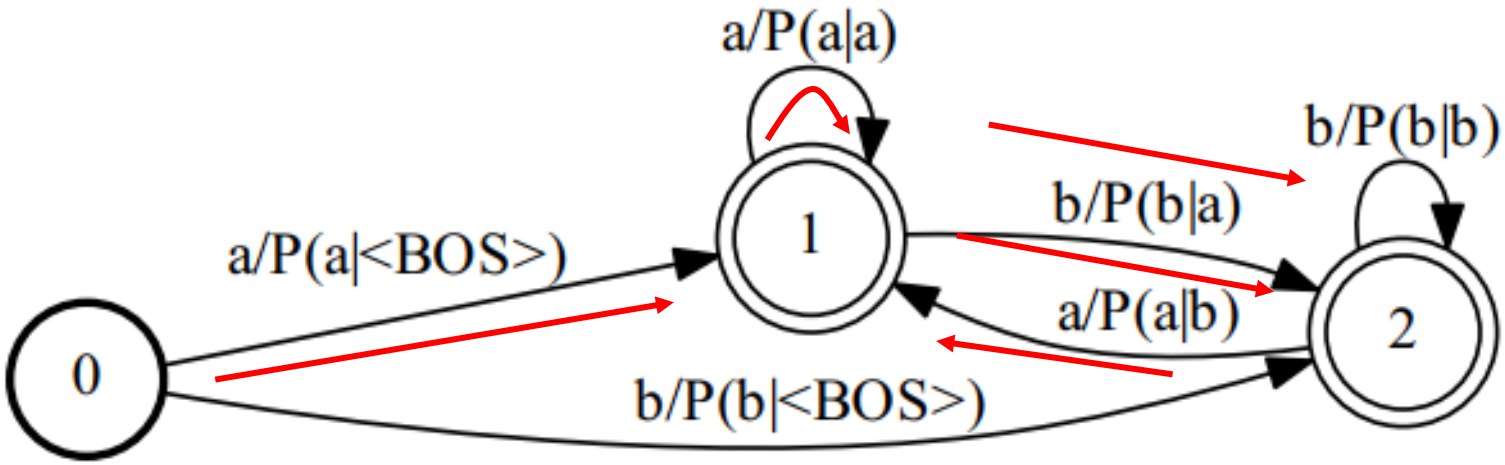
语言模型的自动机表示

- 例子：考虑句子a b a a b



语言模型的自动机表示

- 例子：考虑句子a b a a b



$$P(<BOS>, a, b, a, a, b, <EOS>)$$
$$= P(<BOS>)P(a|<BOS>)P(b|a)P(a|b)P(a|a)P(b|a)P(<EOS>|b)$$

语言模型的自动机表示

- 考虑带回退的语言模型

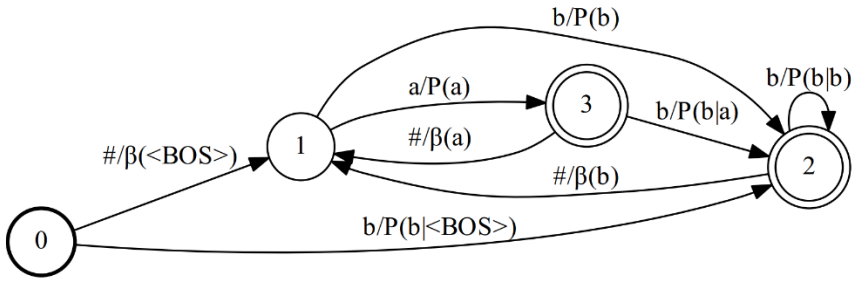
$$\hat{P}(w|h) = \begin{cases} P(w|h) \\ \beta P(w|h') \end{cases}$$

- 加入语言模型的概率如下表

$P(a)$	$\beta(a)$	$P(b\|a)$	$P(<EOS>\|a)$
$P(b)$	$\beta(b)$	$P(b\|<BOS>)$	$P(<EOS>\|b)$
$\beta(<BOS>)$	$P(b\|b)$		

语言模型的自动机表示

- 可以构建这样的自动机



- #为特殊符号，保证确定性。
- 对于序列 a b a a b，概率为

$$P(<BOS>,\ a,\ b,\ a,\ a,\ b,\ <EOS>) =$$

$$P(<BOS>)\beta(<BOS>)P(a)P(b|a)\beta(b)P(a)\beta(a)P(a)\beta(a)P(a)P(b|a)P(<EOS>|b)$$

发音词典的自动机表示

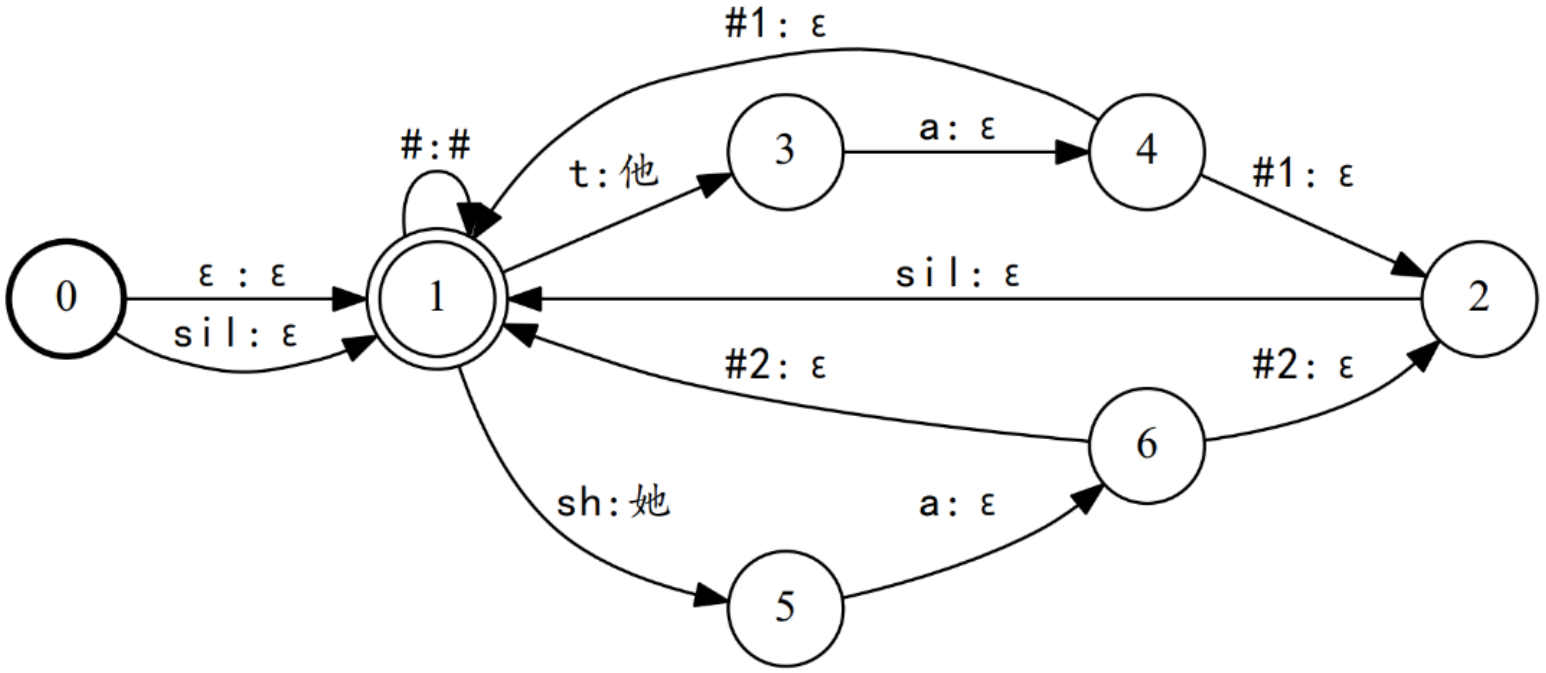
- 发音词典是音素序列和词的对应关系。
- 下图是一个简单的发音词典

他	t a #1
她	t a #2

- 其中 #1 和 #2 是去混淆符号，防止生成的WFST不是确定的。

发音词典的自动机表示

- 可以构建如下的转换器



- 其中，SIL表示前后可能的静音。

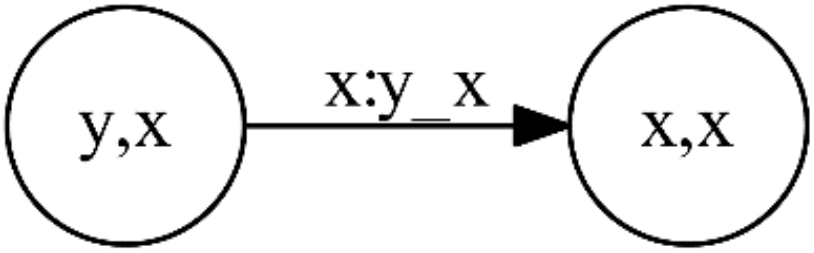
上下文相关的音素

- 符号phone/left_right来表示一个上下文相关的三音素。
- 比如，以u的一个三音子模型为例，其左边是h，右边是a，则此三音子模型可以表示为u/h_a。这个三音子模型可能发生在“华”的发音当中。
- 假设两个音素x，y，则它们能组成的三音素有

x/x_x	y/x_x	x/x_y	y/x_y
x/y_x	y/y_x	y/x_y	y/y_y

上下文相关的音素

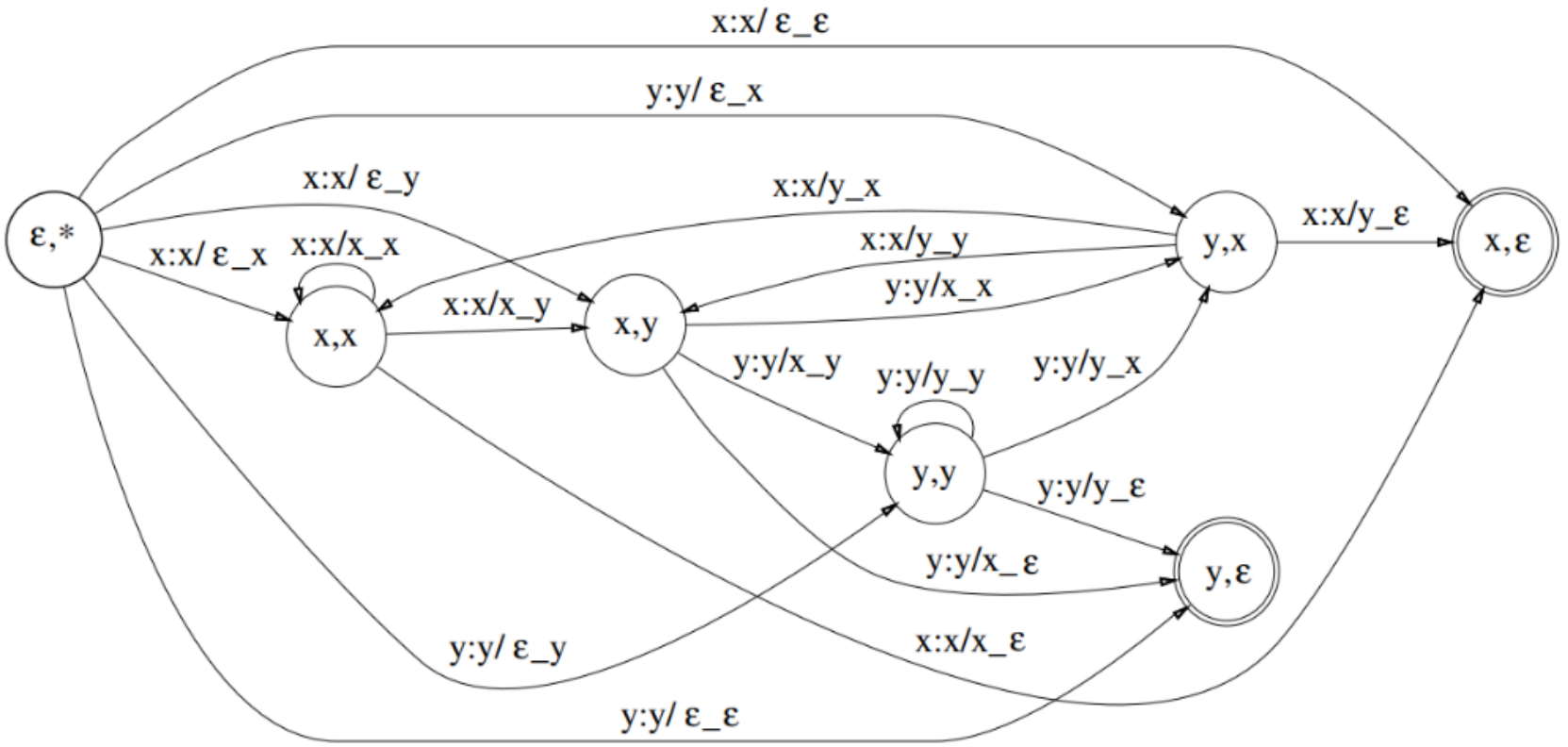
- 比如单音素序列 x y x x y，可以扩展为
  x/ε_y y/x_x, x/y_x, x/x_y, y/x_ε
- 对于第三个位置的x来说，左边是 y，右边是x。将三音素作为边。对 x/y_x，可以看坐是(y,x)到(x,x)状态的转移，如下图。



- 对每一个三音素可以构造相应的转移

上下文相关的音素

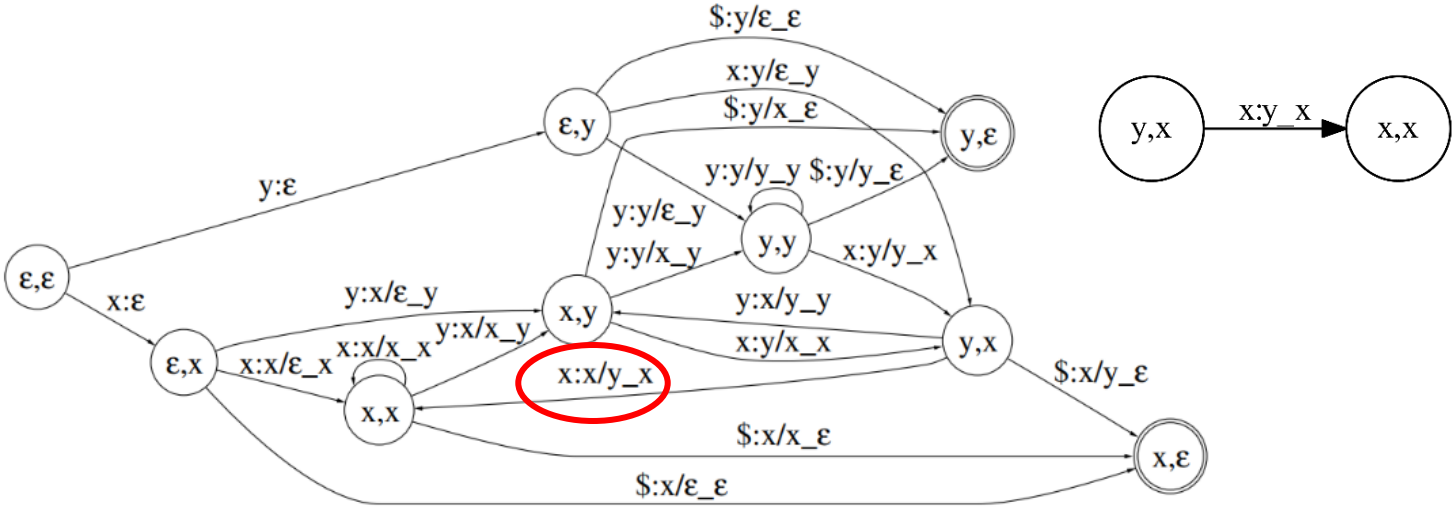
- 比如单音素序列 x y x x y，可以扩展为
  x/ε_y y/x_x, x/y_x, x/x_y, y/x_ε
- 完整的WFST如下



然而，直接构造的WFST不是确定化的。

上下文相关的音素

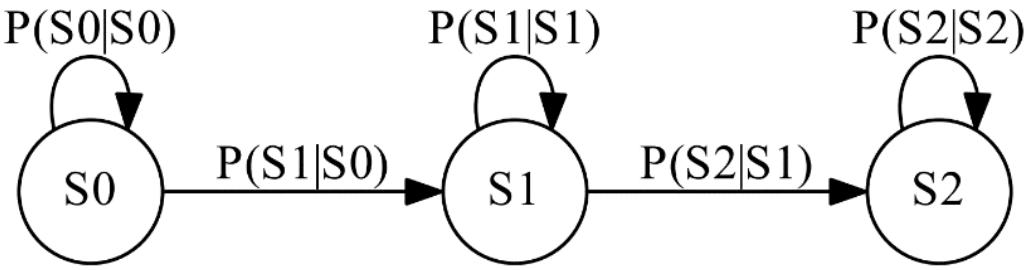
- 原始的WFST以三音素phone/left_right中的phone作为输入。现在将输入符号后推移”一个，将中的right标为输入符号。
- 在开头添加从空状态(ε, ε)到第一个音素的边。在最后添加$作为最后right占位符。



- 将音素到三音子转换器上每一个转移的输入和输出交换，即可得到三音素到音素的转换器。

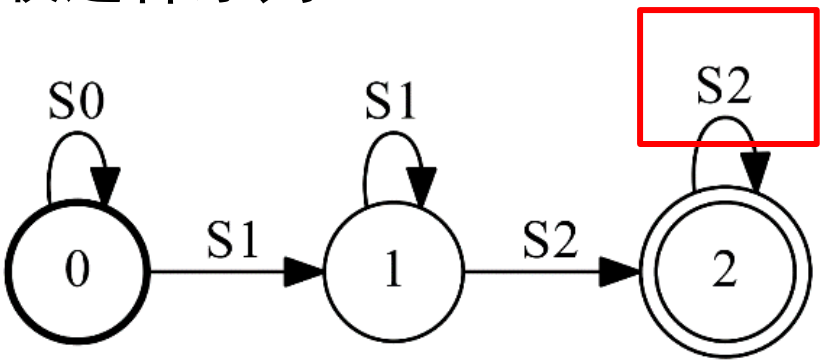
隐马尔可夫模型

- 三状态链式隐马尔可夫模型（bakis topology）对一个三音子建模



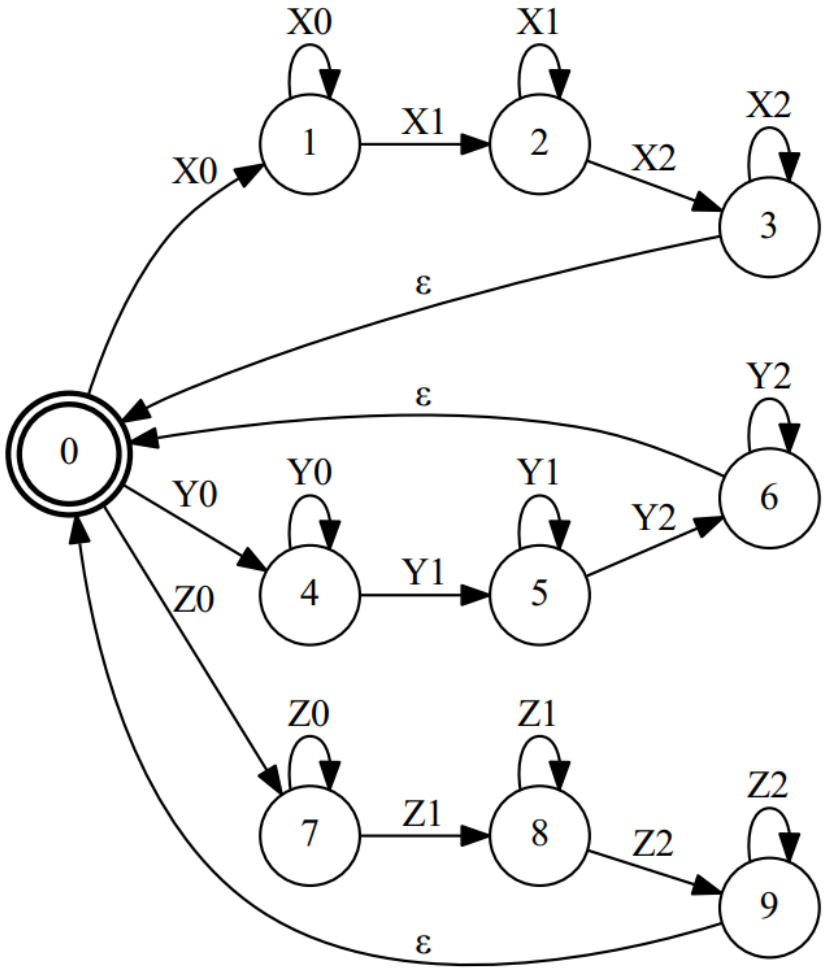
- 边上的概率表示状态转移概率。
- 这种拓扑可以生成s0 s0 s0 s0 s1 s1 s2 s2 s2 这样的序列。
- 构建WFST接收这种序列。

注意区别于HMM拓扑。这个图的边代表的是HMM状态。



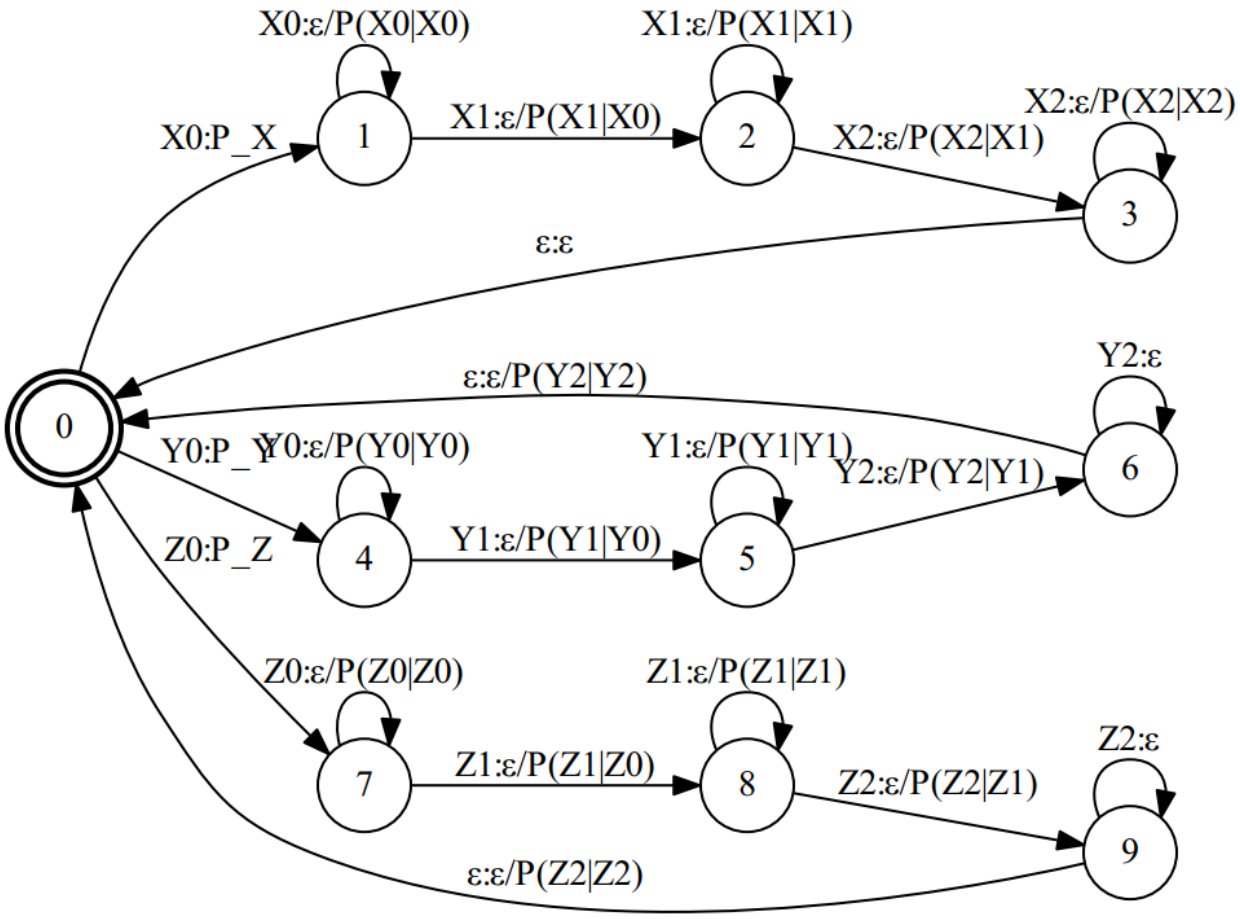
隐马尔可夫模型

- 将不同的隐马尔可夫模型串起来



隐马尔可夫模型

- 构建隐马尔可夫状态到三音素的转换器



完整解码图的构建

$$N = \pi_\varepsilon \min\left(\det\left(H \circ \det\left(C \circ \det\left(L \circ G\right)\right)\right)\right)$$

- H代表隐马尔可夫模型
- C代表上下文相关音素模型
- L代表发音词典
- G代表语言模型
- HCLG往往规模庞大，150是小时标注文本生成的实验用的HCLG有百万节点，五百万边。商用系统可以达到几十亿节点。

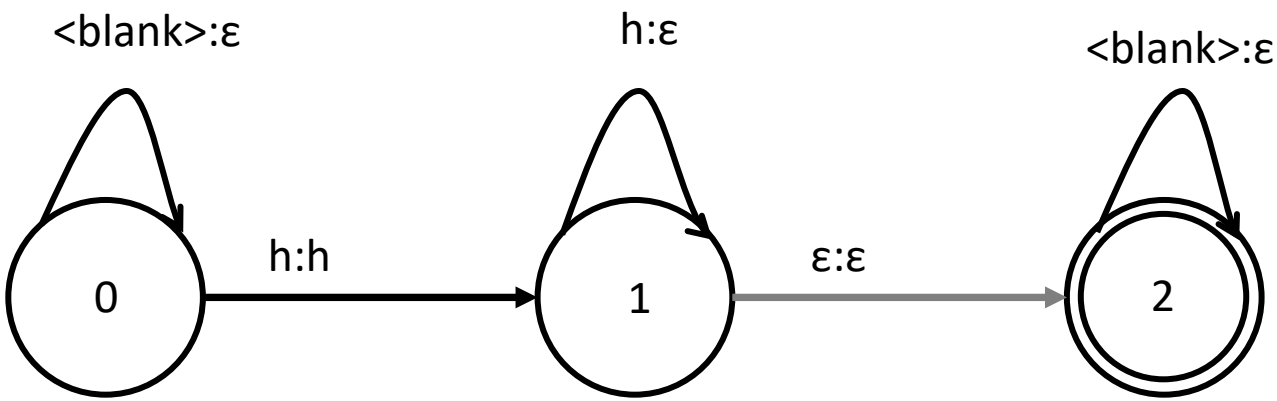
CTC的解码图构建

- 与HMM相比，CTC可以可以直接建模音素，所以不需要H和C。
- CTC解码的关键在于，如何将CTC序列（带blank和重复）转化为标注序列？

c c _ a _ _ t  c a t

T的构建

- 我们可以构建这样的WFST，来将CTC符号序列转换为标注序列。
- 如对于音素 h，可以构建这样的WFST
- 将CTC序列 _ _ _ _ h h h_ _ _ 输入进WFST，看看结果是什么？



TLG的构建

- 将T，L，G合并起来，就可以构建出CTC的解码网络。

$$S = T \circ min(det(L \circ G))$$

提纲

- 解码空间构建
  - WFST回顾
  - 解码图的构建
- 解码搜索
  - 维特比搜索
  - 束搜索
  - 解码实例
- 端到端语音识别模型
  - 为什么要进行端到端语音识别
  - 循环神经网络转换器
  - 基于注意力机制的端到端

搜索技术

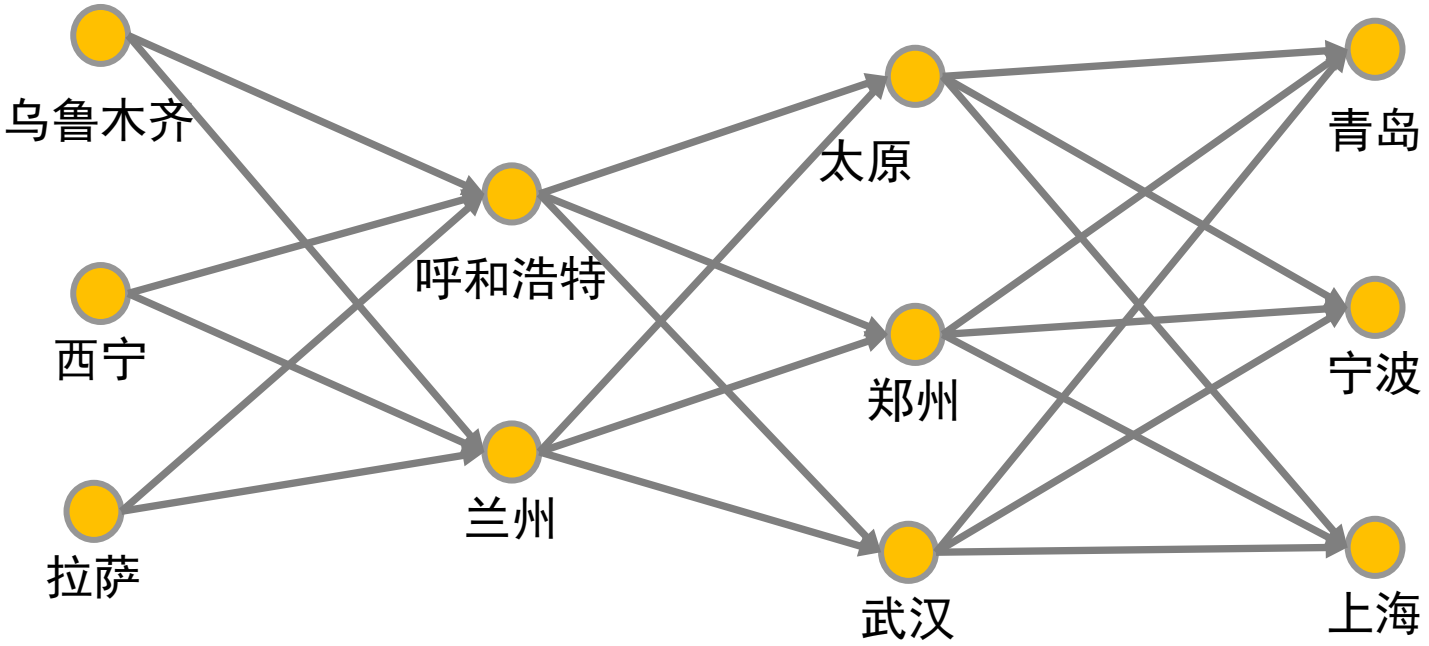
- 构建出解码图以后，我们就可以根据声学分数和解码图上的分数，寻找出概率最大的词序列。
- 这个寻找的过程，称为解码。
- 由于解码图往往十分庞大（商用解码图可以达到数十GB大小，包含数亿个状态，数十亿个转移），求得全局最优路径不现实，往往采用的是启发式算法。

搜索技术

- 图搜索中常用的启发式算法有两种，一种是基于深度优先原则的A*搜索算法，一种是基于广度优先原则的束搜索（Beam Search）。
- 现代语音识别系统中最常用的是束搜索算法。
- 首先介绍维特比（Viterbi）算法的动态规划思想，然后介绍束搜索的基本思想，最后介绍语音识别解码器中的令牌传递模型。

维特比搜索

- 考虑旅游路径规划



- 节点表示城市，边表示城市之间的距离，考虑用三天时间从最左边的城市（西）走到最右边（东），每天只能从一个城市走到另一个城市，不能走两个城市，问如何规划路线最短？

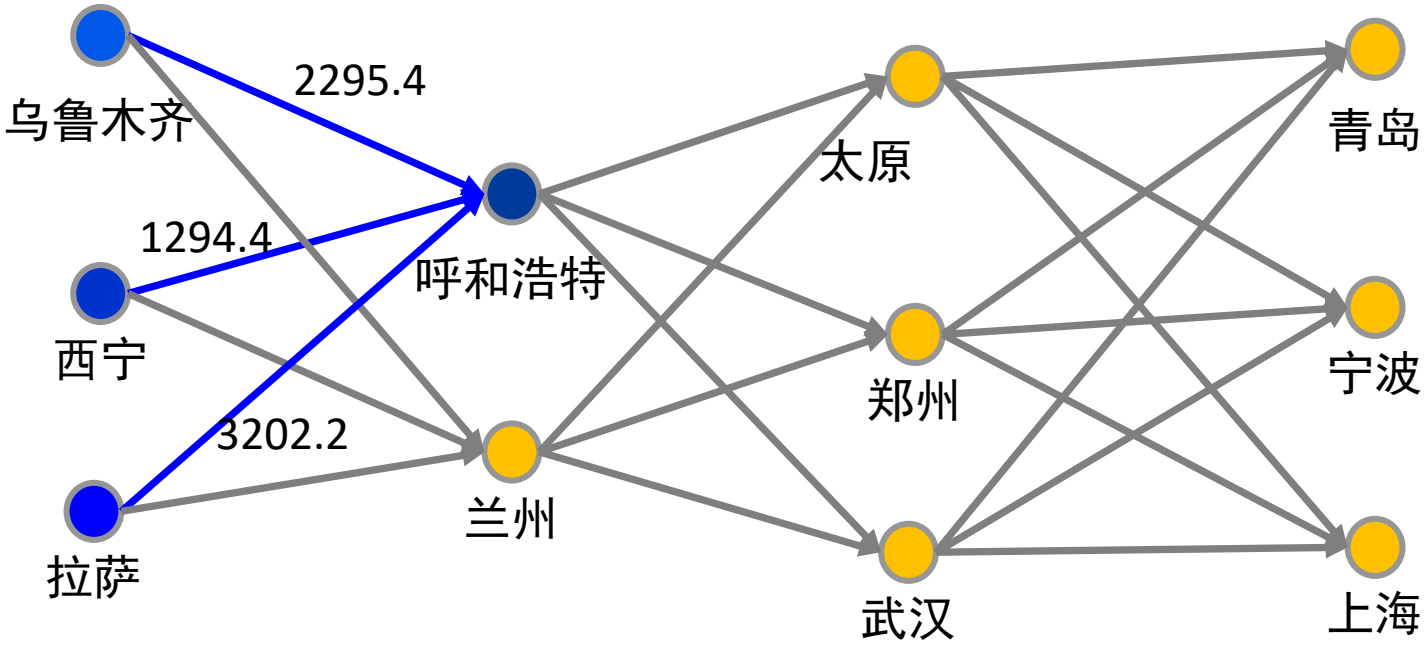
维特比搜索

- 格图符合这样的性质：当前状态仅依赖前一步决策与前一个状态，与其它状态无关。
- 维特比算法就是一种格图上求最短路径的动态规划算法，其状态定义为每一步的累积路径长度。

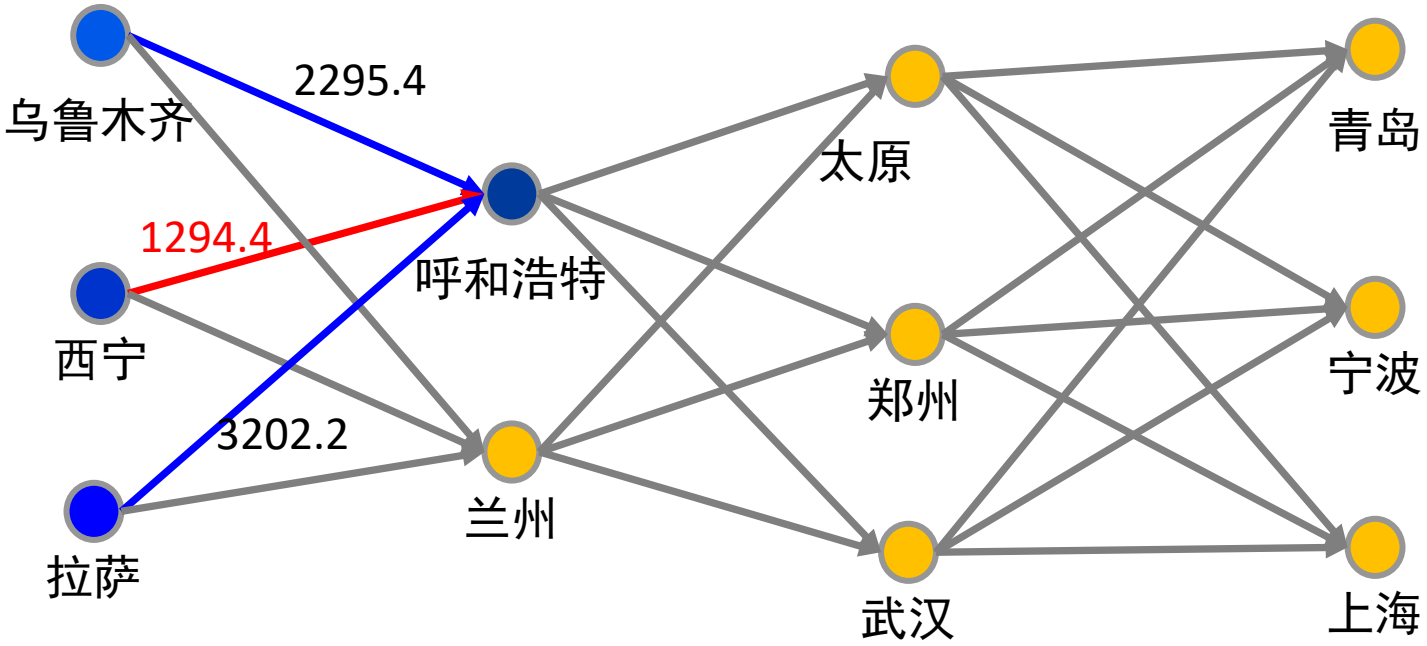
令牌传递

- 令牌传递由Steve Young提出，是一个语音识别解码算法的一般化的模型，其也经常用于手写文字识别等。
- 令牌（Token）是一个数据结构，其保存了在解码图上搜索的时候，当前决策步下的累积分数，对应的决策，以及此决策步下对应的加权有限状态转换器上的输入符号和输出符号等信息。
- 解码的过程可以看成是在图上一步一步将令牌向前传递的过程。
- 传递到最后，找到累积分数最优的令牌，再对此令牌对应的路径进行回溯，就可以得到最优路径了。

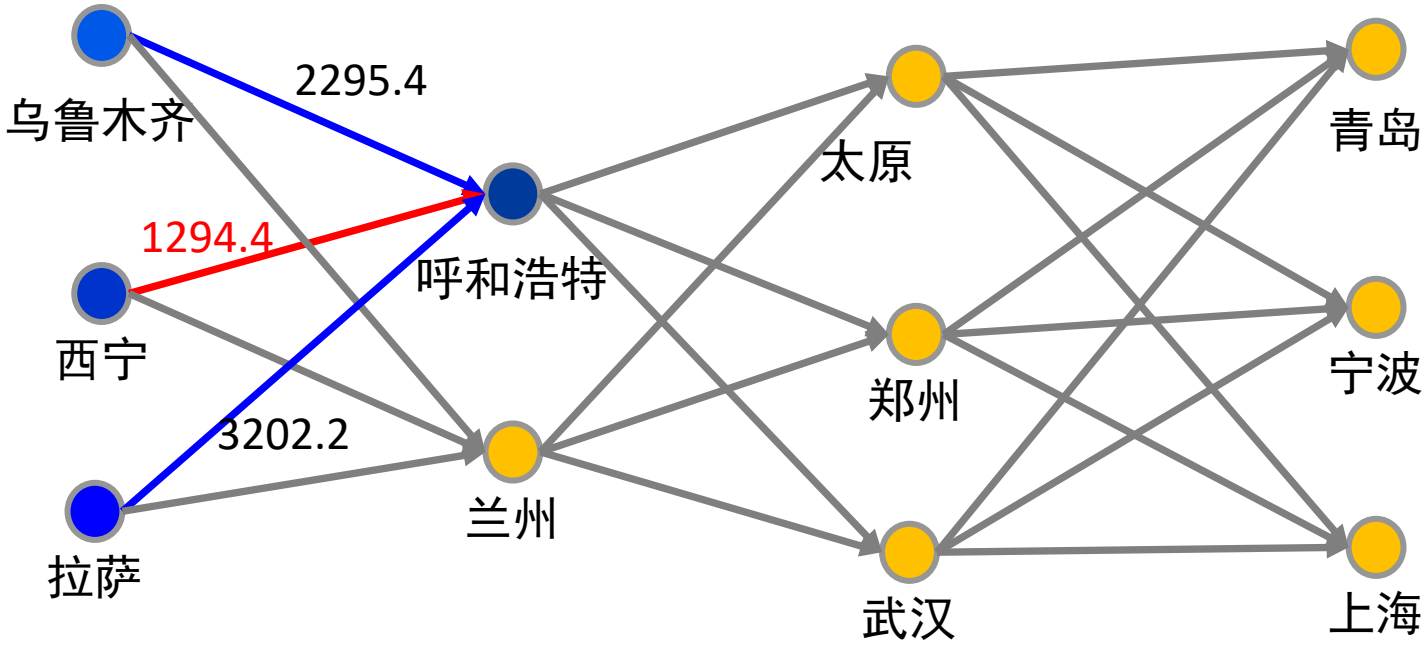
例子



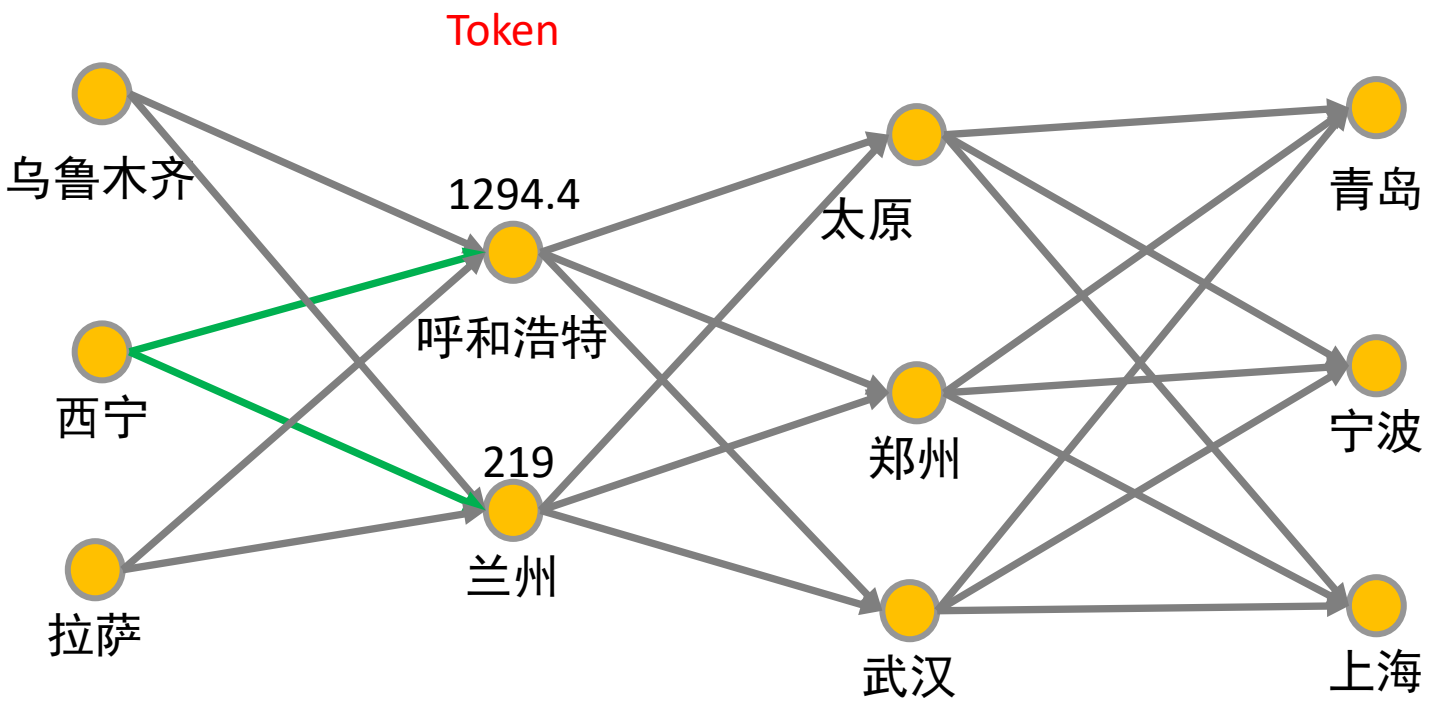
例子



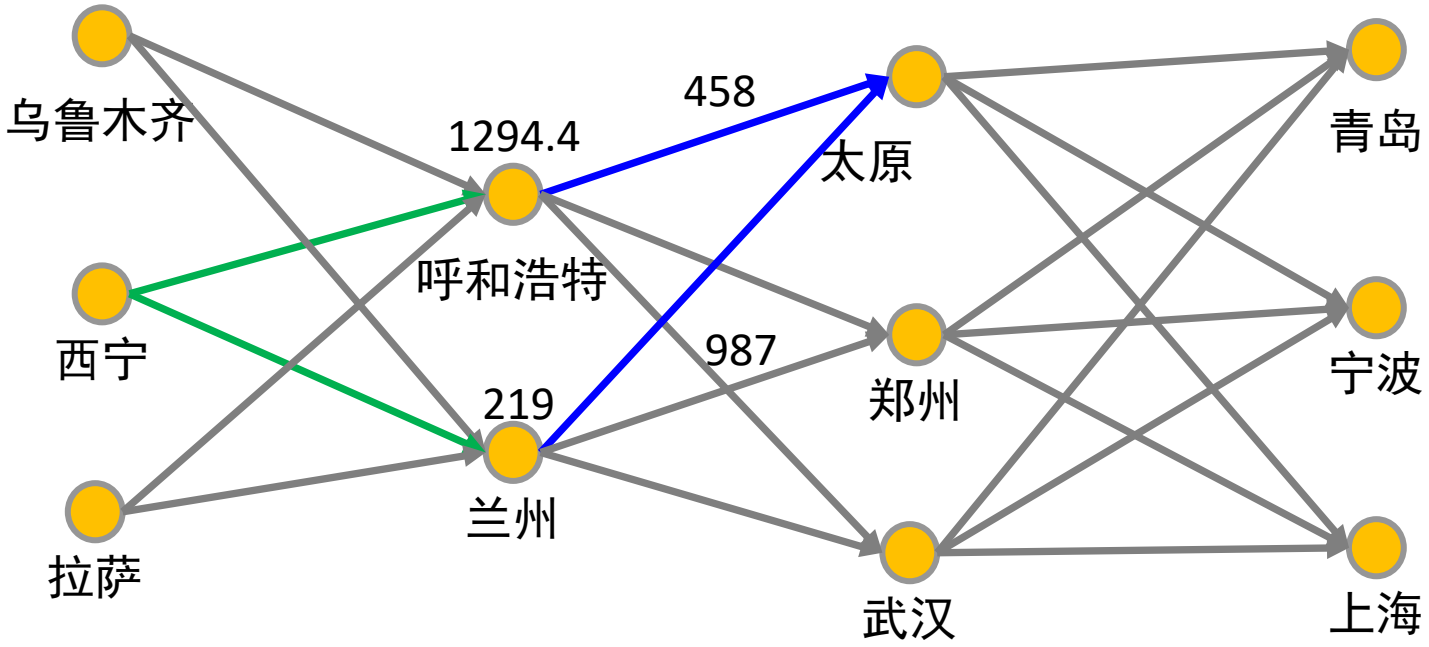
例子



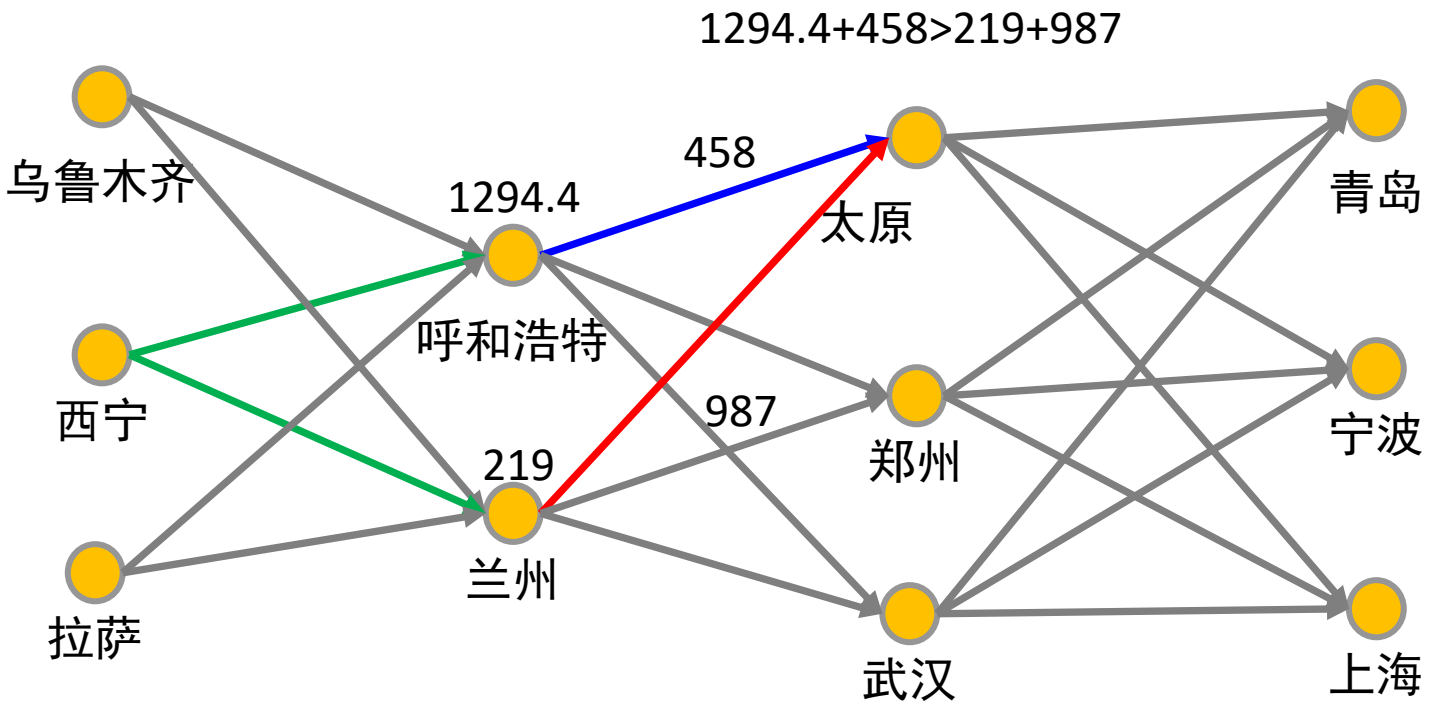
例子



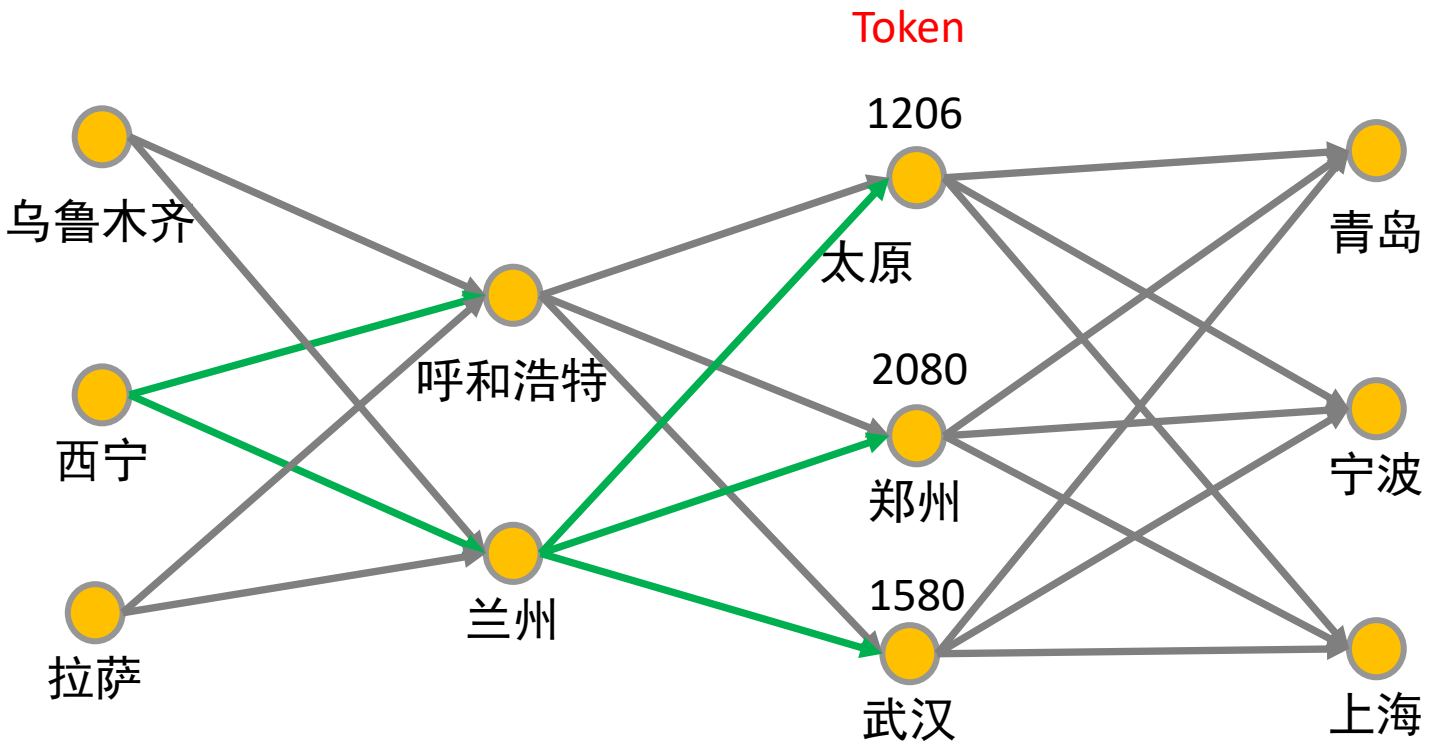
例子



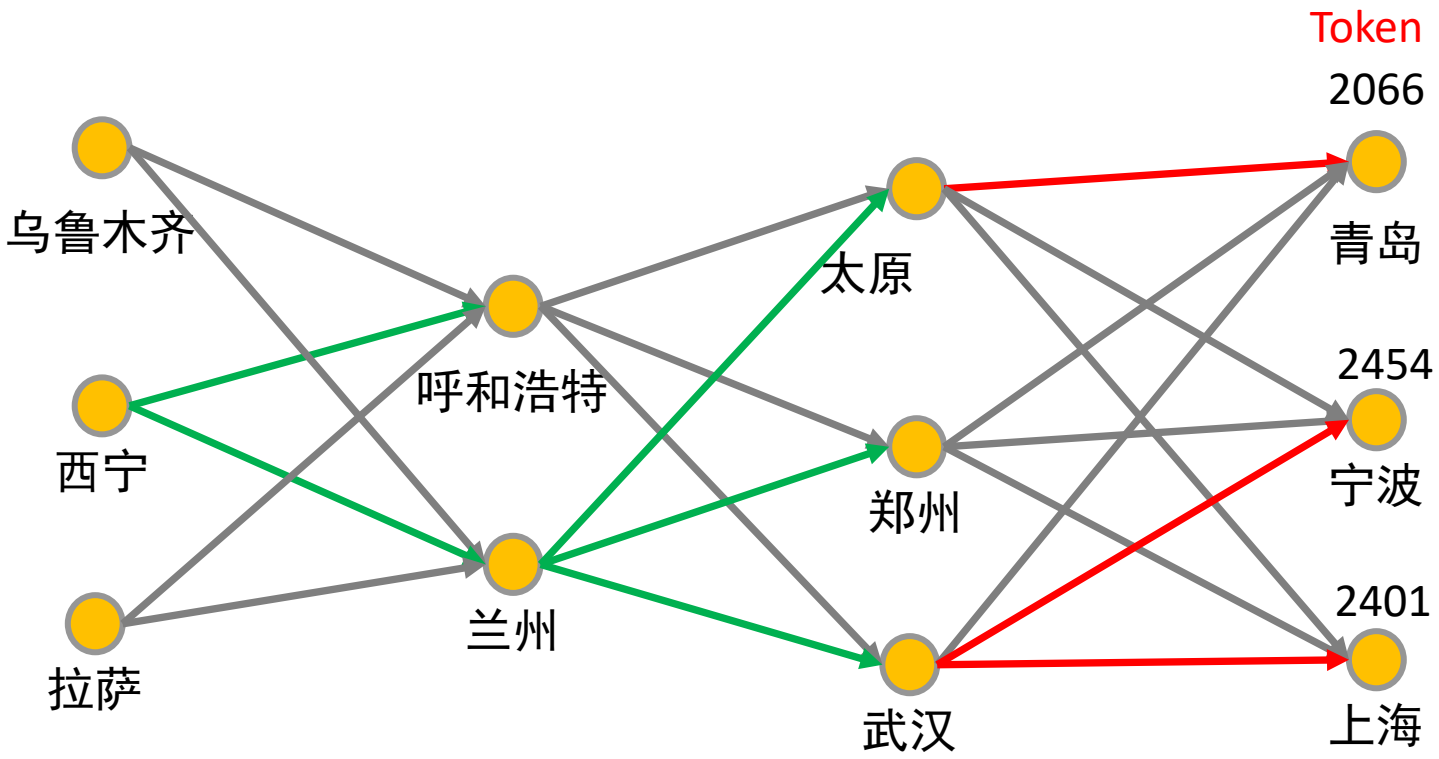
例子


1294.4+458>219+987
乌鲁木齐
西宁
拉萨
1294.4
呼和浩特
219
兰州
458
太原
987
郑州
武汉
青岛
宁波
上海

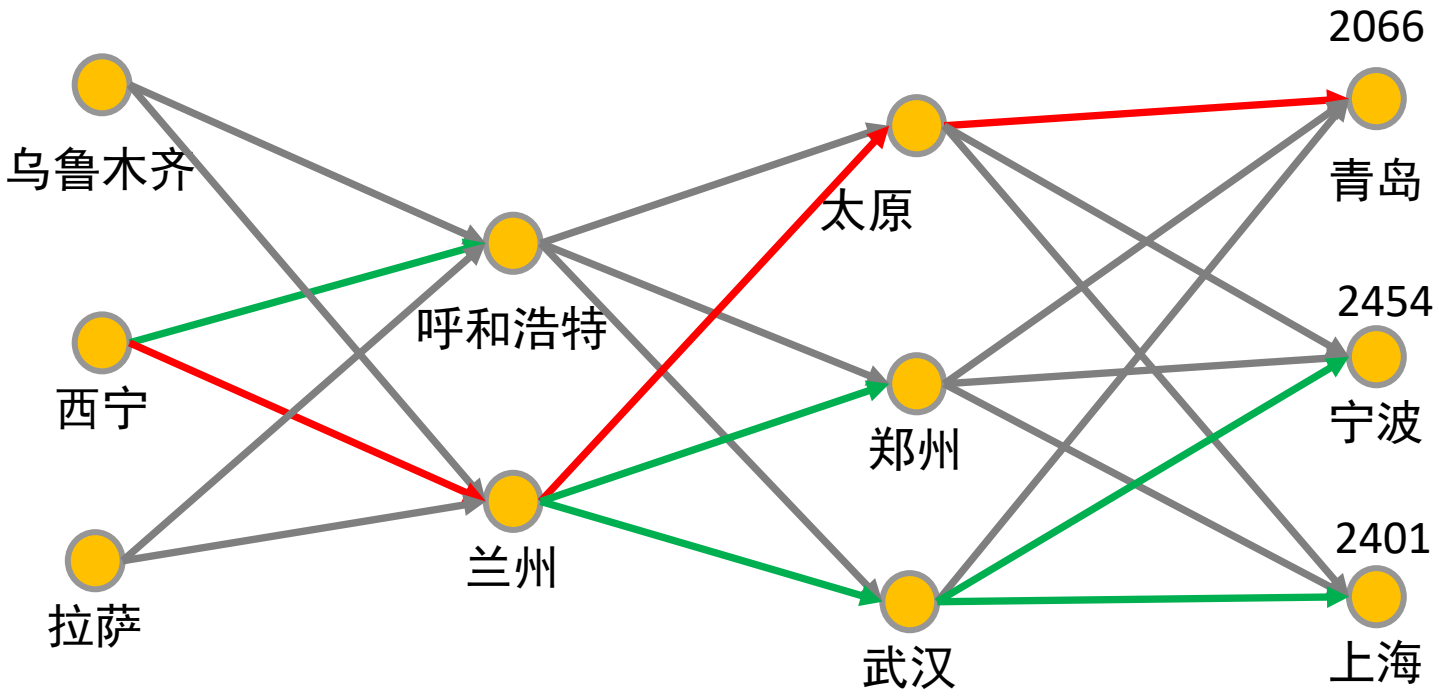
例子



例子



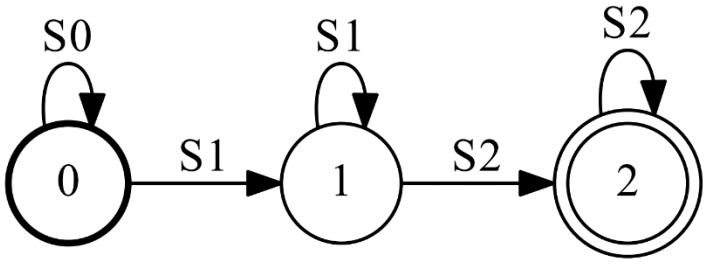
例子



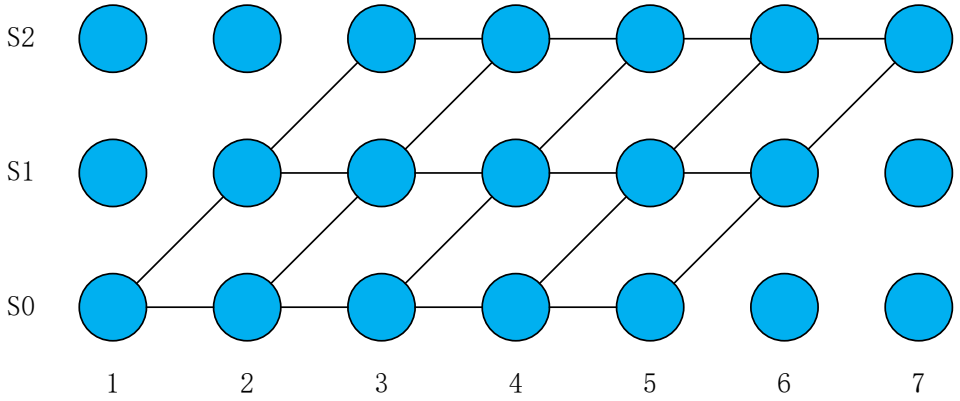
传递到最后，找到累积分数最优的令牌，再对此令牌对应的路径进行回溯，就可以得到最优路径了。

HMM自动机上的维特比

- 考虑前面介绍的HMM自动机



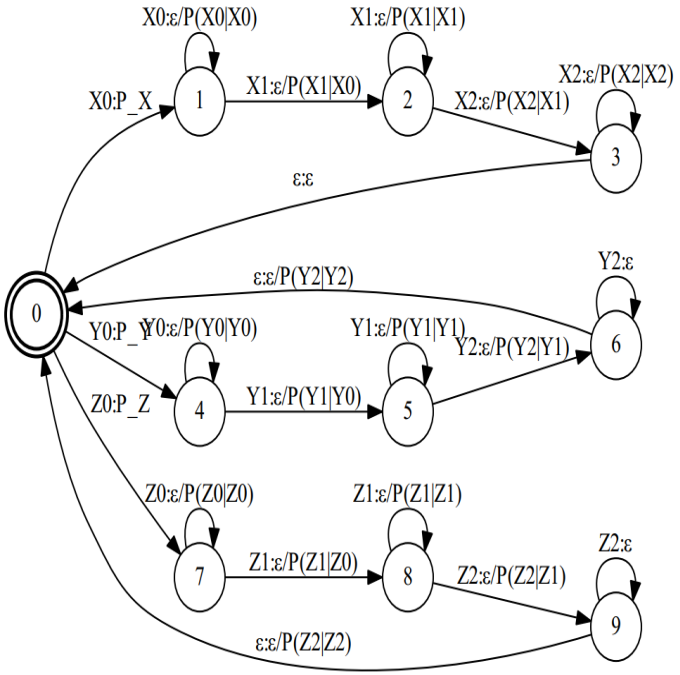
- 我们可以构建对应的格图



- 据维特比算法，在每一个状态节点记录权重累积权重最小者，连接起来，就可以得到权重最小的隐马尔可夫状态序列。

语音识别的解码

- 在语音识别当中，解码图HCLG(TLG)也是自动机，可以类似地构造格图进行维特比解码。
- 假设给定一个T帧的语音序列，我们就可以构建一个T步的格图，格图上节点之间的连接由加权有限状态转换器HCLG (TLG)上的拓扑结构决定，格图上的权重则是由声学模型和HCLG (TLG)上转移的权重加和得到。
- 根据维特比搜索算法，可以得到这T帧对应的HCLG (TLG)上的权重（概率分数)最大的转移的序列，这个转移序列的输出符号序列就是对应的解码出的词序列。



提纲

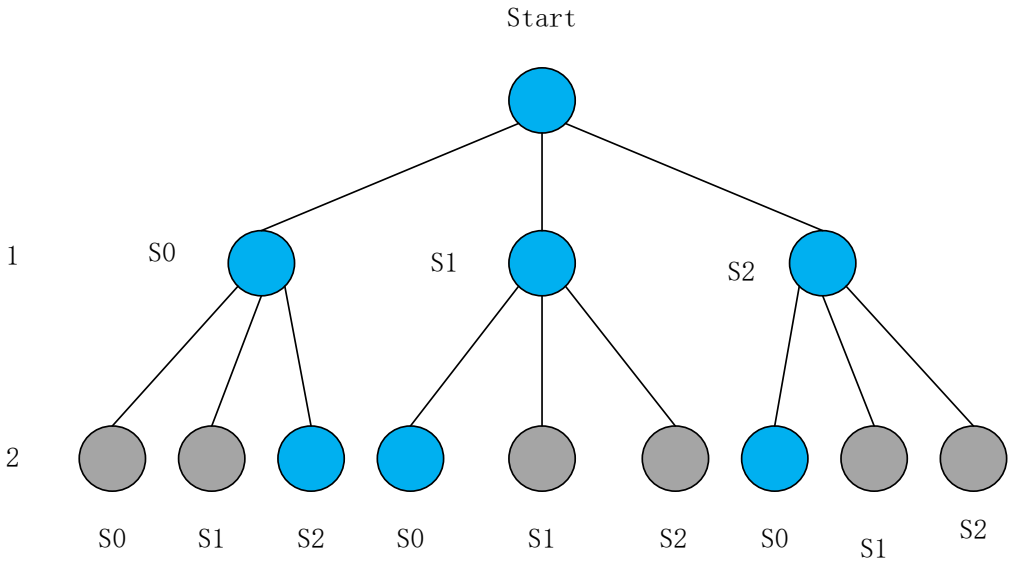
- 解码空间构建
  - WFST回顾
  - 解码图的构建
- 解码搜索
  - 维特比搜索
  - 束搜索
  - 解码实例
- 端到端语音识别模型
  - 为什么要进行端到端语音识别
  - 循环神经网络转换器
  - 基于注意力机制的端到端

束搜索

- HCLG解码图往往规模十分庞大，直接利用维特比算法进行全搜索是不可能的。需要利用启发式算法找到近似解。
- 束搜索是一种基于宽度优先原则的启发式图搜索技术。
- 但与宽度优先搜索不同，束搜索在每一步仅会考虑某种指标下优先的一些决策，并以此进行扩展，而其它部分“剪枝”掉。

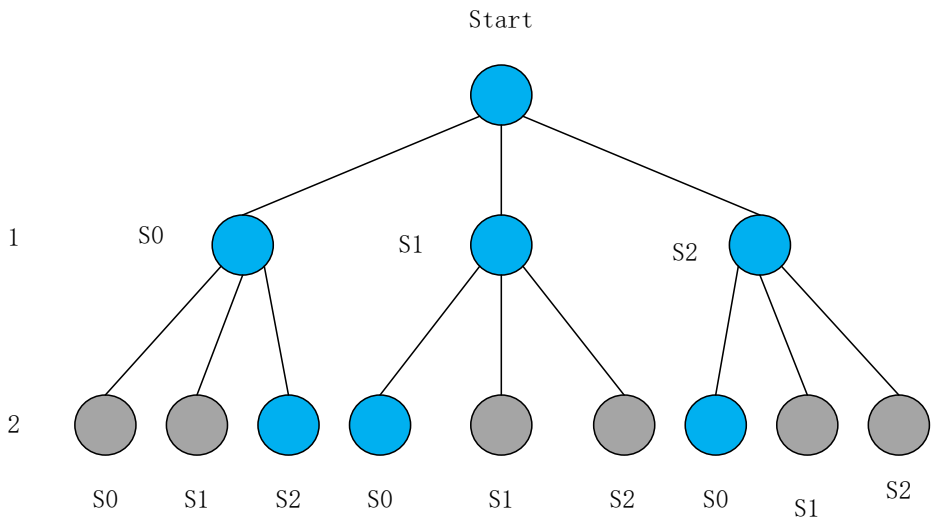
束搜索

- 我们将格图上的搜索表示为搜索树



- 树上蓝色的节点表示搜索过程中保留下来的前几个节点，灰色节点表示被“剪”掉的节点。

束搜索



- 根据宽度优先原则，此树在每一步将会以当前步激活的节点为基础，考虑下一步可能做出的所有决策，计算每一个决策的得分，然后根据这个得分，进行下一步决策，扩展优先级高的节点，剪除其它节点，形成“束”（即最开始只有一个起始节点，随着决策步数增多，树的叶子节点越来越多，形状是一个“束”）；决策到最后一步的时候，得到最后得分最高的路径。

束搜索

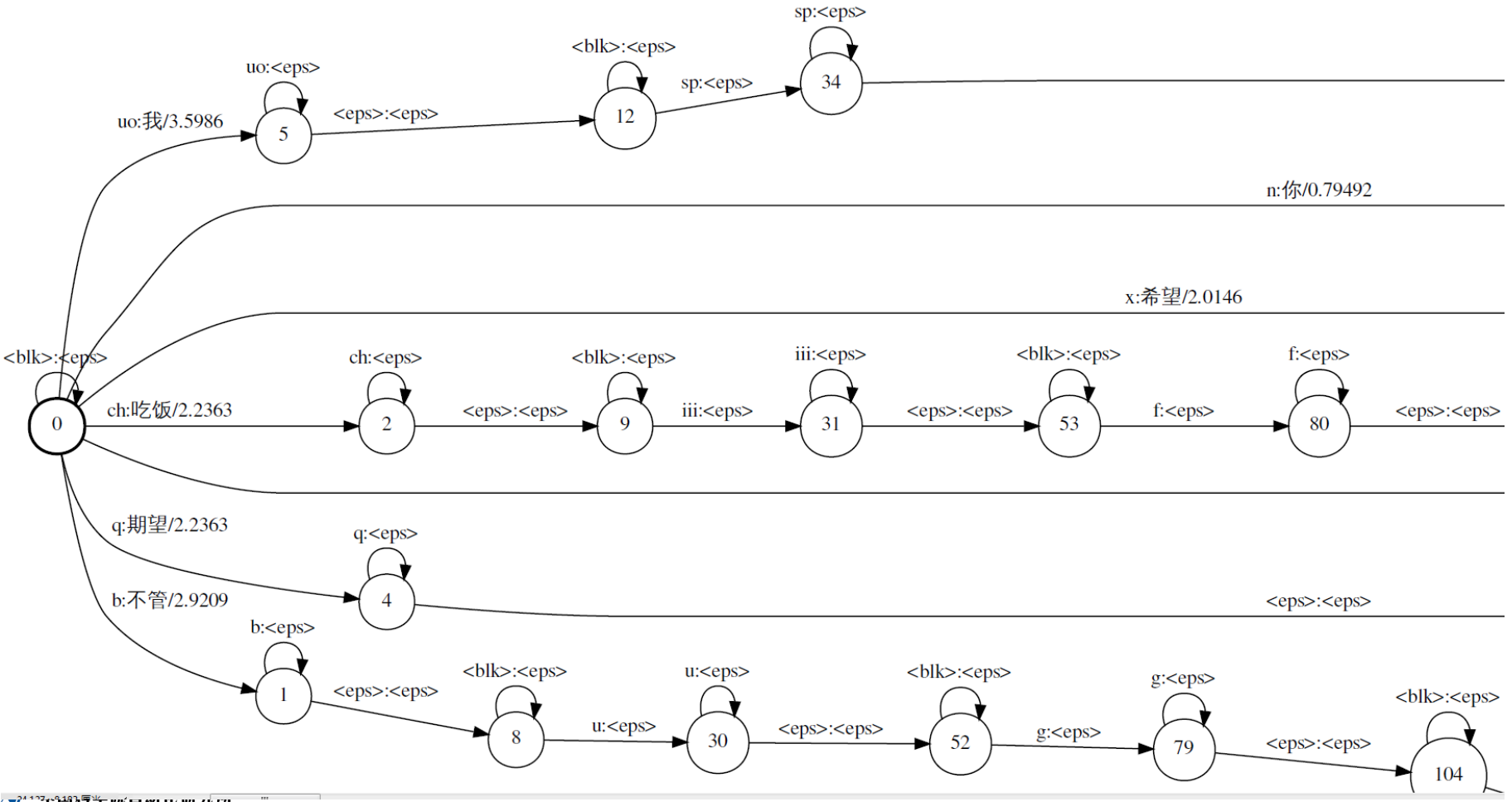
- 剪枝的准则的设计则有多种，常用的有两种方法
- 1. 每一步保留前k个累积负对数概率值最小的节点
- 2. 以累积负对数概率值最小的节点（最优路径）为基准，设定一个分数阈值，如果一个节点超出最优节点的得分大于这个阈值，就将其剪除。

提纲

- 解码空间构建
  - WFST回顾
  - 解码图的构建
- 解码搜索
  - 维特比搜索
  - 束搜索
  - 解码实例
- 端到端语音识别模型
  - 为什么要进行端到端语音识别
  - 循环神经网络转换器
  - 基于注意力机制的端到端

解码实例：CTC解码

- 声学模型: CTC声学模型
- 语言模型: 3-gram



解码实例：CTC解码

声学分数

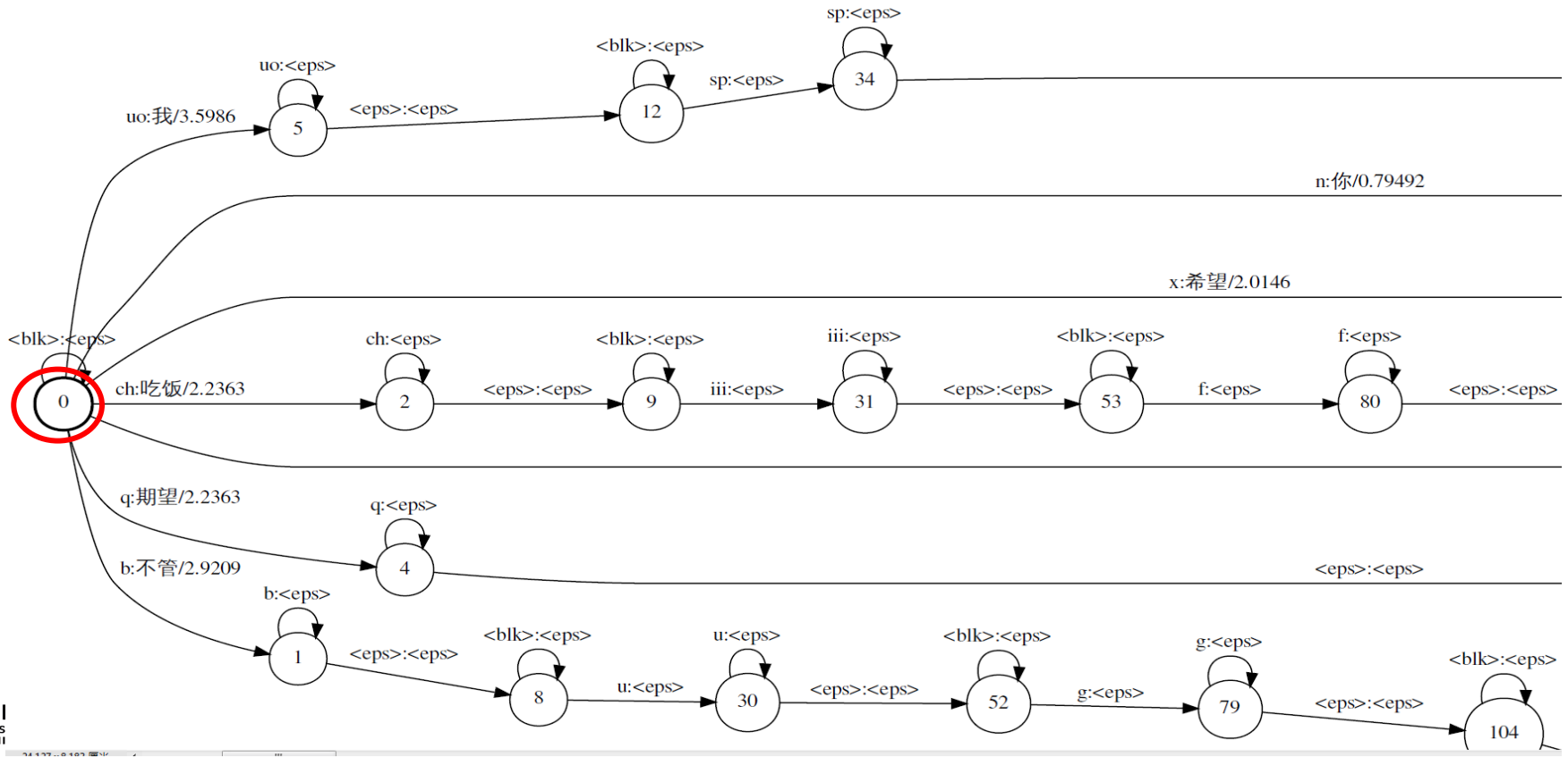
1	phn/frame	1	2	3	4	5	6
2	blk	-0.007180121	-0.999769	-10.51062	-12.58604	-8.497993	-0.0008461
3	spn	-11.99832	-8.265933	-10.83303	-9.999769	-10.61947	-10.61947
4	a	-8.709741	-8.739963	-9.919142	-8.265933	-12.58604	-12.58604
5	ai	-10.80694	-10.1651	-9.728236	-8.739963	-9.999769	-10.51062
6	an	-9.564792	-9.841268	-7.68909	-10.1651	-0.0024375	-10.83303
7	ao	-10.6956	-7.224031	-8.833096	-9.841268	-9.564792	-9.919142
8	b	-10.3361	-11.75618	-7.224031	-7.224031	-10.6956	-9.728236
9	c	-7.136627	-0.0111345	-11.75618	-11.75618	-10.3361	-7.68909
10	ch	-10.27794	-0.001145542	-7.68909	-10.83303	-7.136627	-8.833096
11	d	-9.320827	-10.1651	-8.833096	-9.919142	-10.27794	-10.51062
12	e	-8.497993	-10.83303	-7.224031	-9.728236	-8.265933	-10.83303
13	eng	-10.61947	-9.919142	-11.75618	-7.68909	-8.739963	-9.919142
14	f	-12.58604	-9.728236	-7.68909	-0.00114661	-10.1651	-9.728236
15	g	-9.999769	-7.68909	-8.833096	-7.979339	-10.51062	-7.68909
16	h	-8.265933	-8.833096	-7.224031	-11.16072	-10.83303	-8.265933
17	i	-8.739963	-9.999769	-0.02177932	-8.679531	-9.919142	-8.739963
18	ian	-10.1651	-8.265933	-0.2197932	-10.61947	-9.728236	-10.1651
19	iii	-9.841268	-8.739963	-0.00842456	-9.320827	-10.83303	-9.841268
20	ing	-7.224031	-10.1651	-7.68909	-8.497993	-9.919142	-8.739963
21	k	-11.75618	-9.841268	-8.833096	-10.61947	-9.728236	-10.1651
22	m	-10.51062	-7.224031	-8.497993	-12.58604	-7.68909	-9.841268
23	n	-10.83303	-11.75618	-10.61947	-10.80694	-8.833096	-7.224031
24	q	-9.919142	-0.037184	-12.58604	-9.564792	-8.497993	-11.75618
25	sh	-9.728236	-0.2161466	-9.999769	-10.6956	-10.61947	-10.51062
26	sp	-7.68909	-8.679531	-8.265933	-8.739963	-12.58604	-10.83303
27	t	-8.833096	-10.61947	-7.224031	-10.1651	-9.999769	-9.919142
28	u	-7.979339	-12.58604	-11.75618	-9.841268	-8.265933	-9.728236
29	uan	-11.16072	-9.999769	-7.68909	-8.497993	-0.1024375	-7.68909
30	uang	-8.679531	-8.265933	-10.1651	-10.61947	-8.265933	-8.833096
31	uei	-10.61947	-10.83303	-10.83303	-9.564792	-8.739963	-7.979339
32	uo	-12.58604	-12.58604	-9.919142	-9.320827	-10.51062	-9.841268
33	v	-9.999769	-9.999769	-9.728236	-0.21146616	-10.83303	-7.224031
34	x	-8.265933	-0.12721213	-7.68909	-10.61947	-9.919142	-11.75618
35	z	-10.83303	-10.83303	-8.833096	-12.58604	-9.728236	-10.51062

Step 1

Token

score: 0
sym:

	声学分数	语言分数	累积分数
blk	-0.00718	0	-0.00718
ch	-10.278	2.2363	-8.0417
n	-10.833	0.7949	-10.0381
…			

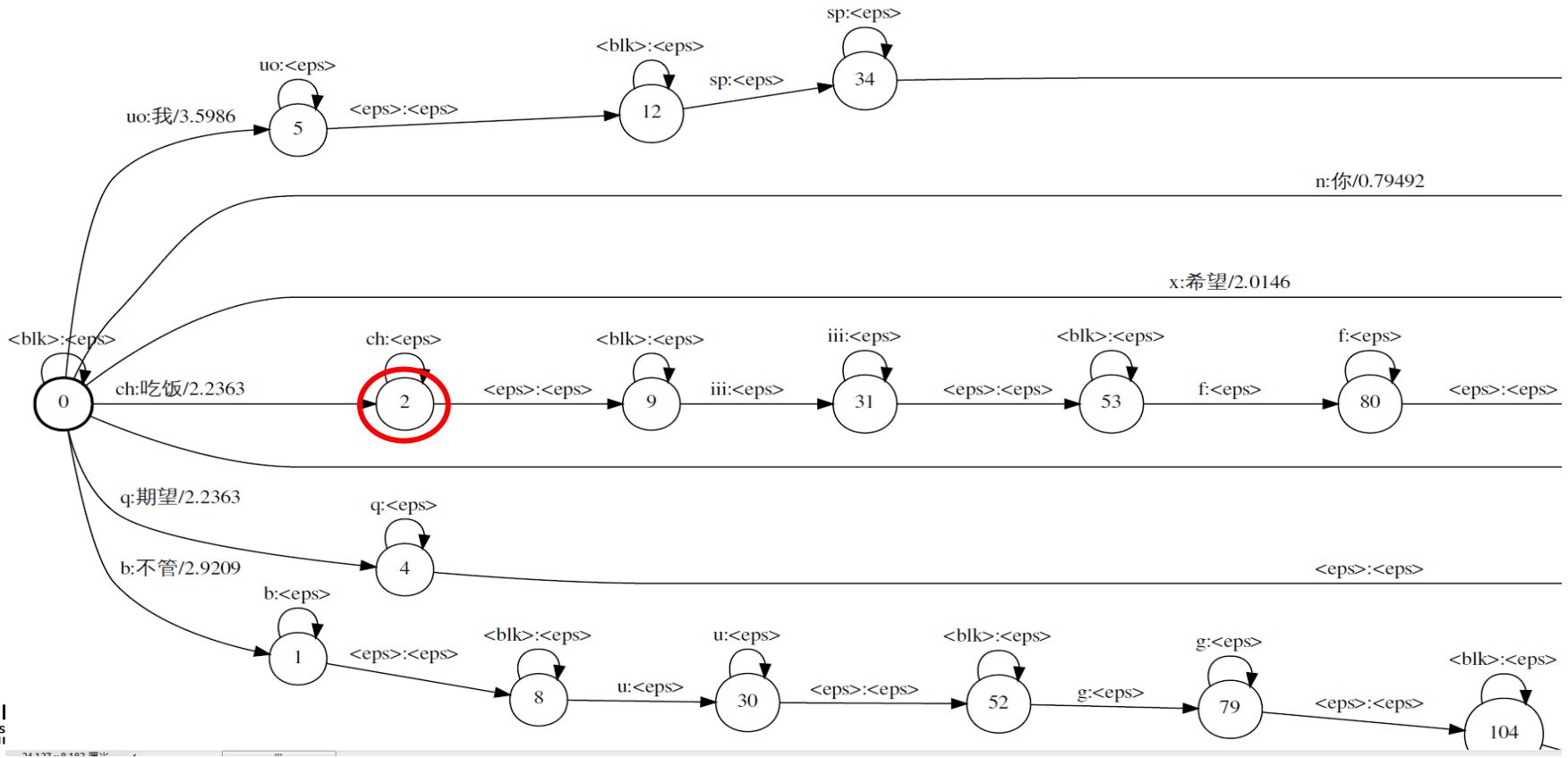


Step 2

Token

score: -0.0718
sym:<blk>

	声学分数	语言分数	累积分数
blk	-0.099	0	-0.1708
ch	-0.001	2.2363	2.1635
n	-11.75	0.7949	-10.96228
…			

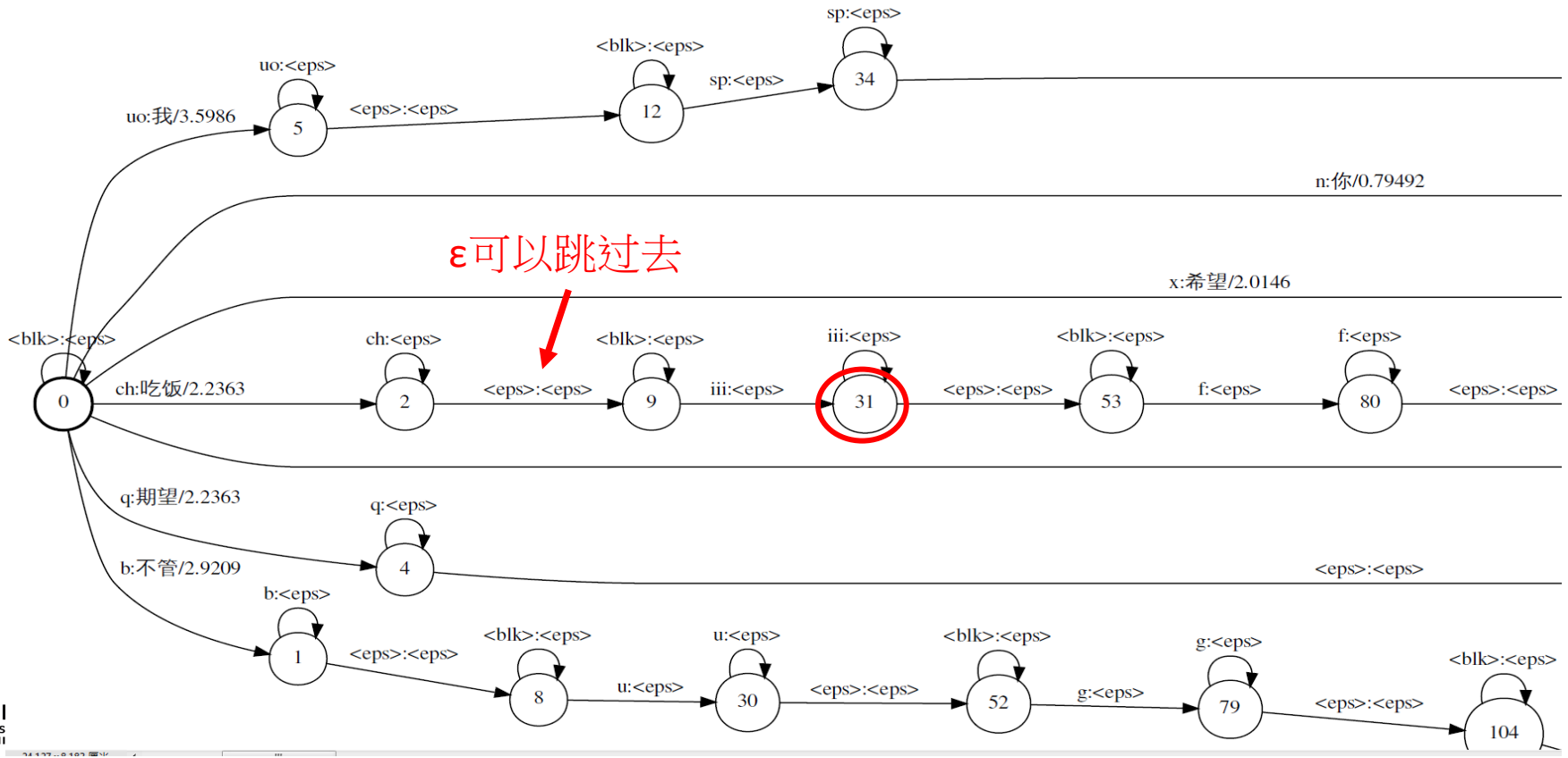


Step 3

Token

score: 2.1635
sym:ch

	声学分数	语言分数	累积分数
blk	-10.51	0	-8.346
ch	-7.69	0	-5.5265
iii	-0.008	0	2.1555
…			

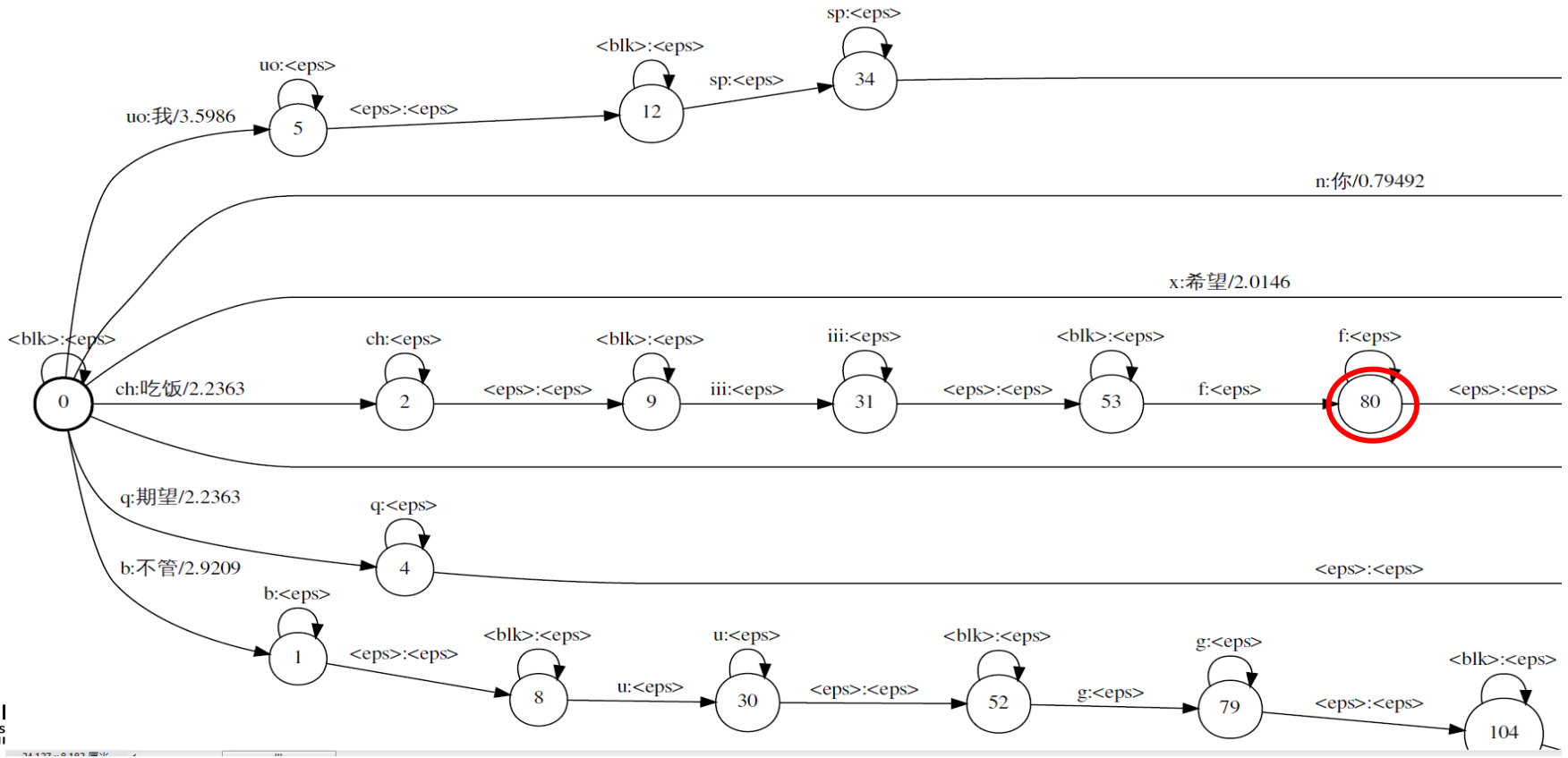


Step 4

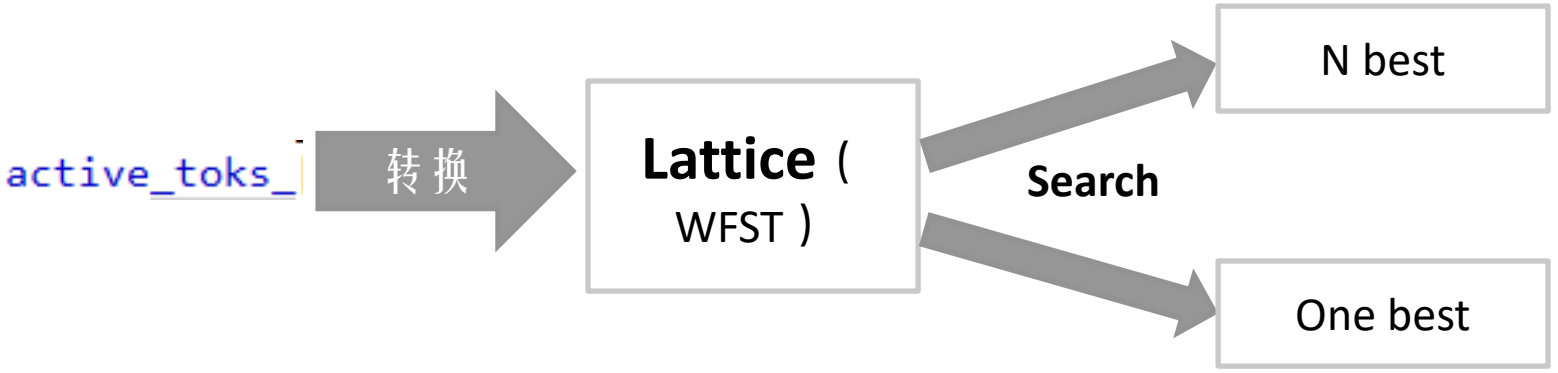
Token

score: 2.1555
sym:iii

	声学分数	语言分数	累积分数
blk	-12.5864	0	-10.4309
f	-0.001	0	2.154
iii	-9.321	0	-7.16
…			



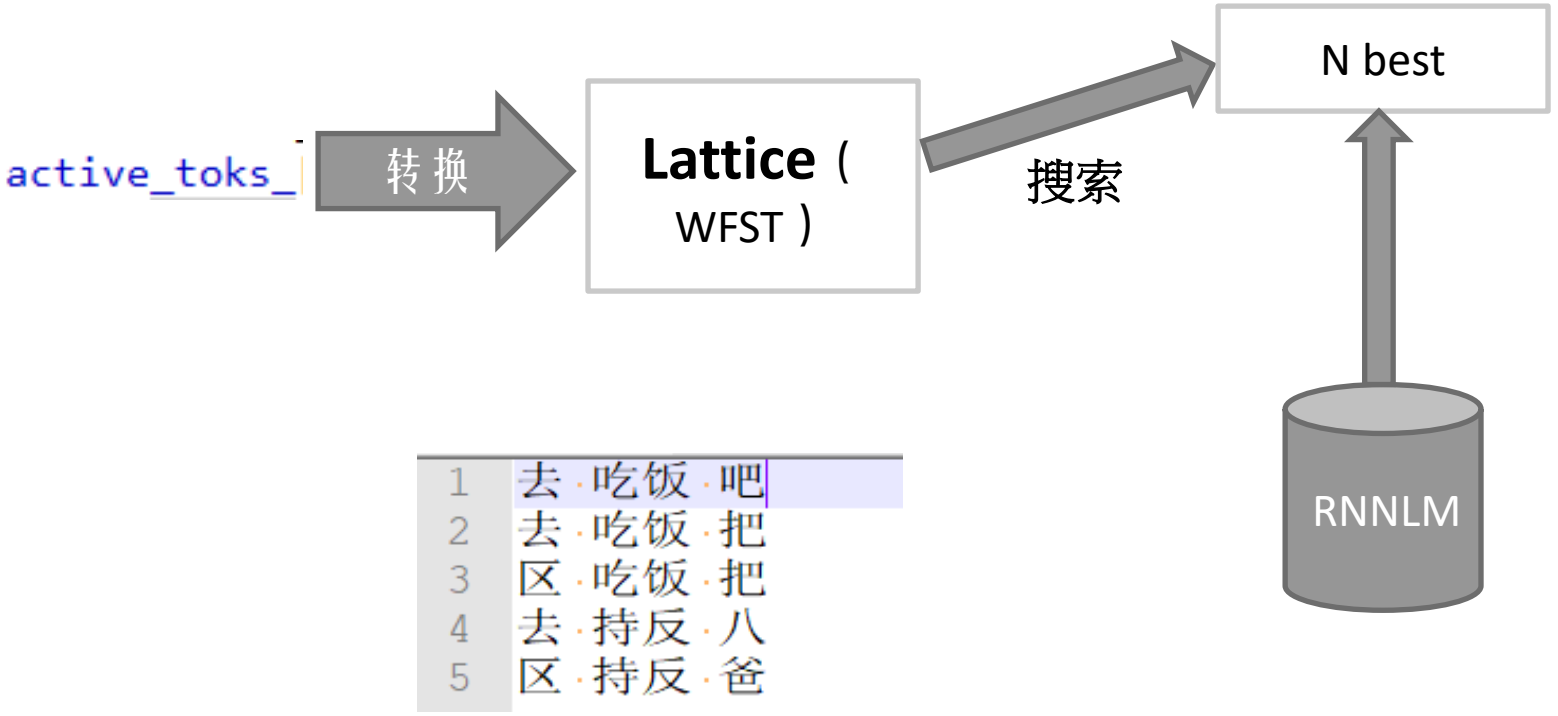
解码结果



1	去·吃饭·把
2	去·吃饭·吧
3	区·吃饭·把
4	去·持反·八
5	区·持反·爸

吃饭 去 吧 ch iii f an q u b a

RNNLM二次重打分


active_toks_
转换
Lattice（
WFST）
搜索
N best
RNNLM
1 去·吃饭·吧
2 去·吃饭·把
3 区·吃饭·把
4 去·持反·八
5 区·持反·爸

提纲

- 解码空间构建
  - WFST回顾
  - 解码图的构建
- 解码搜索
  - 维特比搜索
  - 束搜索
  - 解码实例
- 端到端语音识别模型
  - 为什么要进行端到端语音识别
  - 循环神经网络转换器
  - 基于注意力机制的端到端

为什么要做端到端语音识别模型？

- 传统混合模型包含声学模型、语言模型和发音词典三个模块，很难联合优化，容易造成误差累积。
- 传统混合模型训练神经网络声学模型需要帧级别标注。
- 传统混合模型发音词典需要专家知识，代价高昂。
- 传统混合模型解码图体积庞大。

端到端语音识别系统的优势

端到端系统的优势

端到端联合优化

混合模型的缺点

误差累积

- 端到端语音识别系统声学语言联合优化，避免误差累积。

端到端语音识别系统的优势

端到端系统的优势

混合模型的缺点

序列级优化目标

帧级别标注

- 端到端语音识别系统训练神经网络的时候采用序列级别的优化目标，不需要进行帧级别标注。

端到端语音识别系统的优势

端到端系统的优势

混合模型的缺点

直接建模字符

发音词典

- 端到端语音识别系统直接建模字符，不需要发音词典。

端到端语音识别系统的优势

端到端系统的优势

纯神经网络

混合模型的缺点

体积庞大

- 端到端语音识别系统全部由神经网络构成，不需要构建静态解码图。

端到端语音识别系统的优势

端到端系统的优势

- 端到端联合优化
- 序列级优化目标
- 直接建模字符
- 纯神经网络

混合模型的缺点

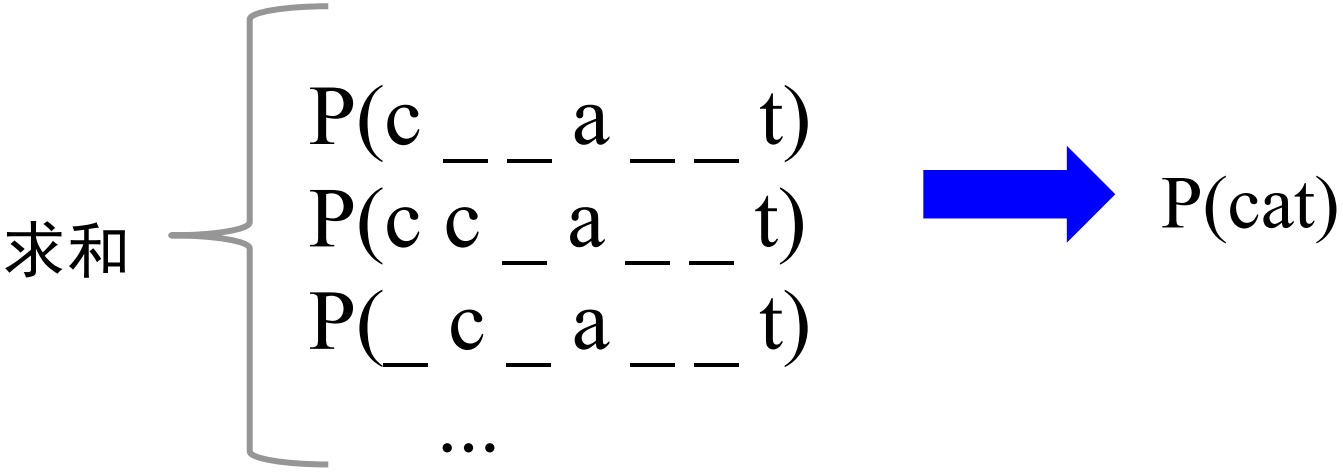
- ✗ 误差累积
- ✗ 帧级别标注
- ✗ 发音词典
- ✗ 体积庞大

主流的端到端语音识别模型

- 联结时序分类模型(CTC，端到端声学模型)
- 循环神经网络语音转写模型(RNN-Transducers)
- 基于注意力机制的序列到序列模型(Attention-based Sequence-to-Sequence)

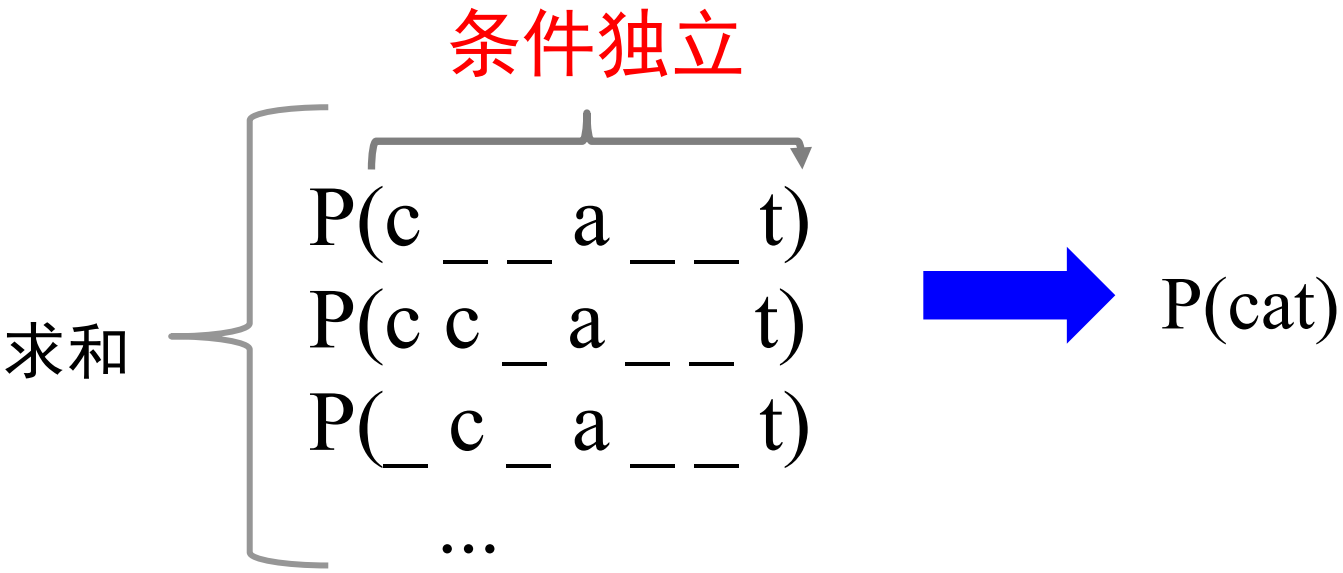
回顾CTC

- 允许标注有重复，并加入blank符号，构成CTC符号序列。
- 假设符号之间条件独立，CTC符号序列概率为连乘。
- 各种可能枚举求和，得到标注序列概率。



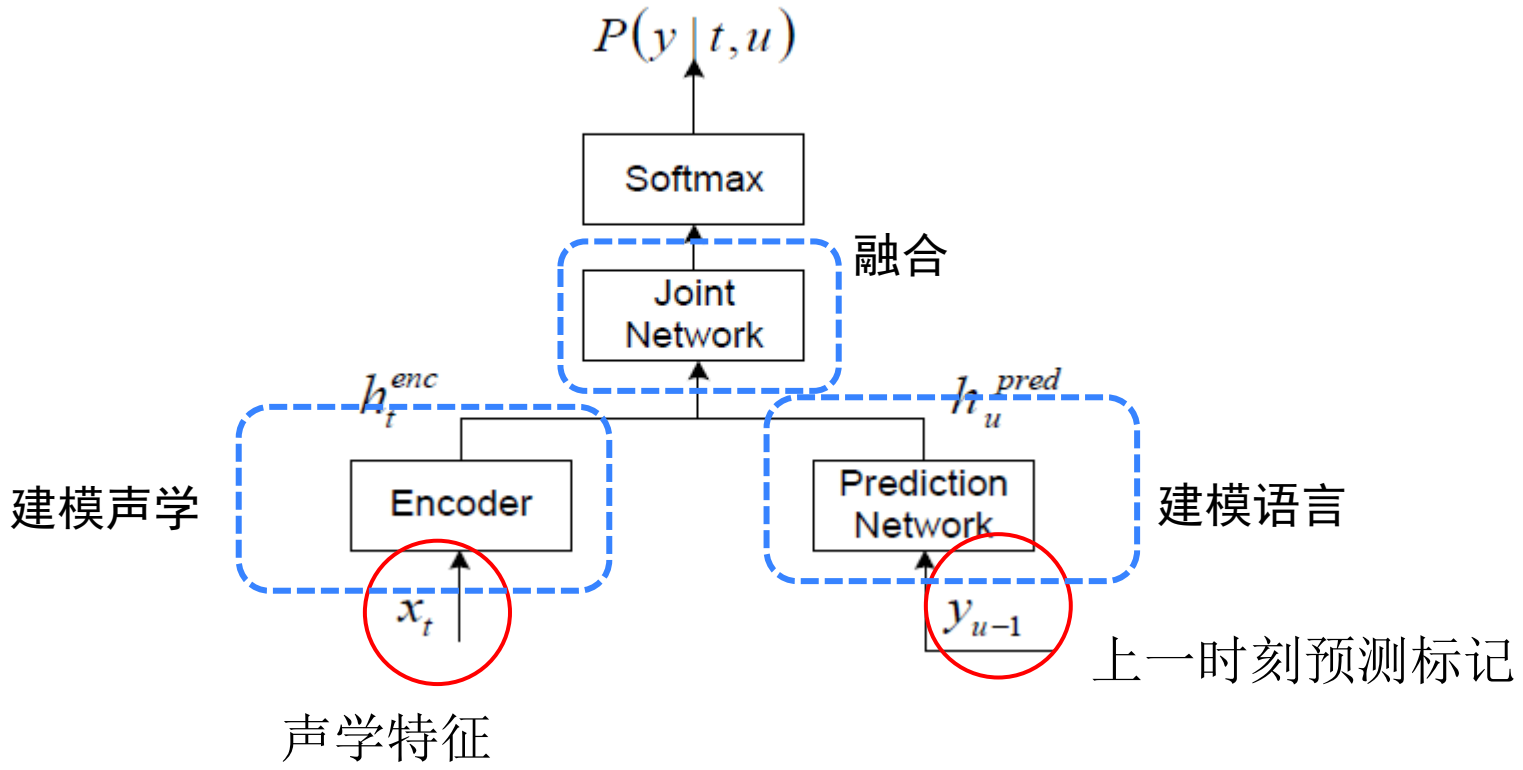
CTC的问题

- 假设符号之间条件独立，对符号之间的依赖关系没有建模，即没有建模语言。



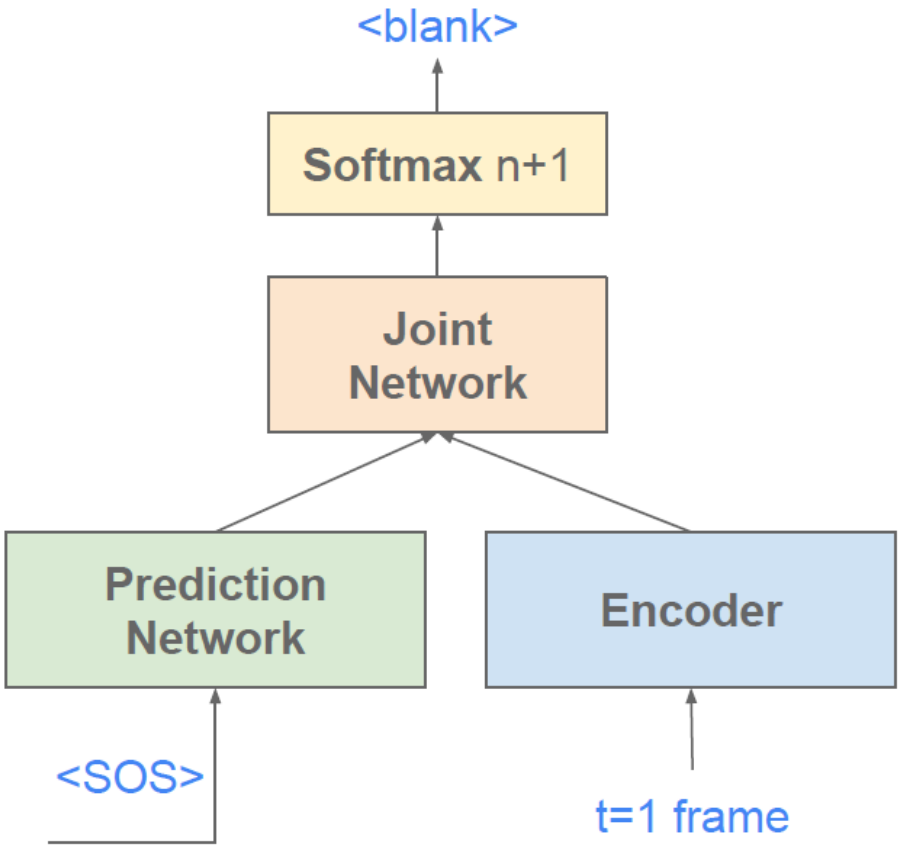
RNN-Transducer

- 利用Prediction network建模符号依赖关系



RNN-T解码示意：第1步

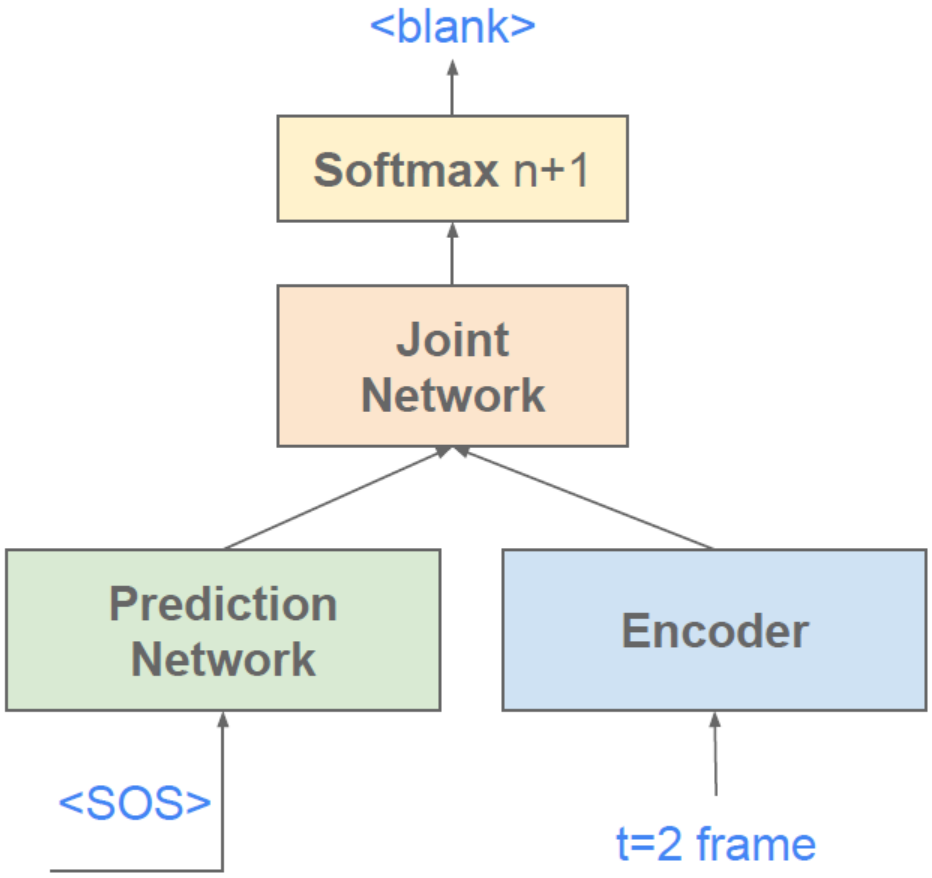
<SOS>



Tara N. Sainath, Towards End-to-End Speech Recognition

RNN-T解码示意：第2步

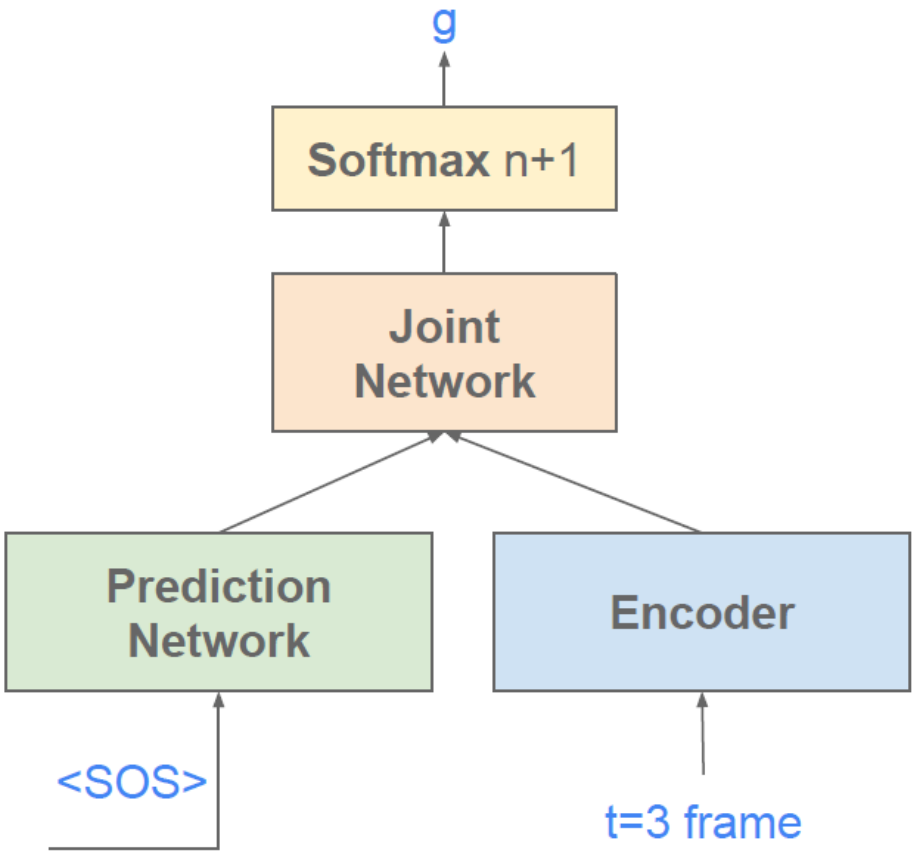
<sos> <blank>



Tara N. Sainath, Towards End-to-End Speech Recognition

RNN-T解码示意：第3步

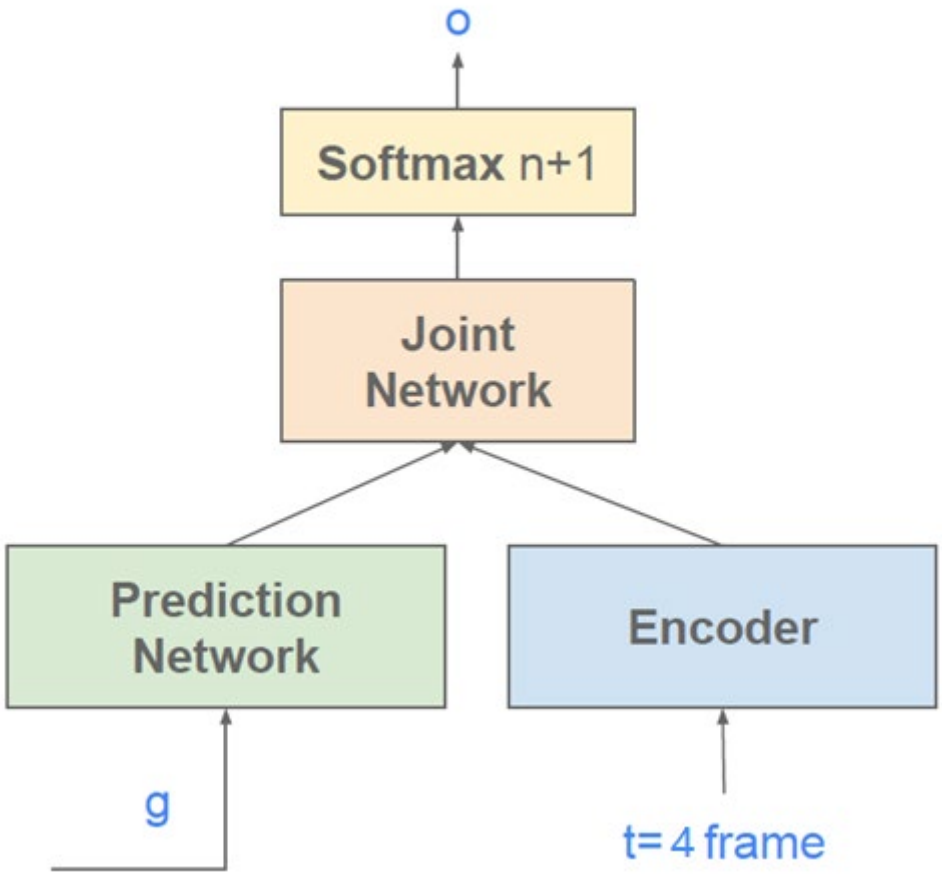
<sos> <blank> g



Tara N. Sainath, Towards End-to-End Speech Recognition

RNN-T解码示意：第4步

<sos> <blank> g o

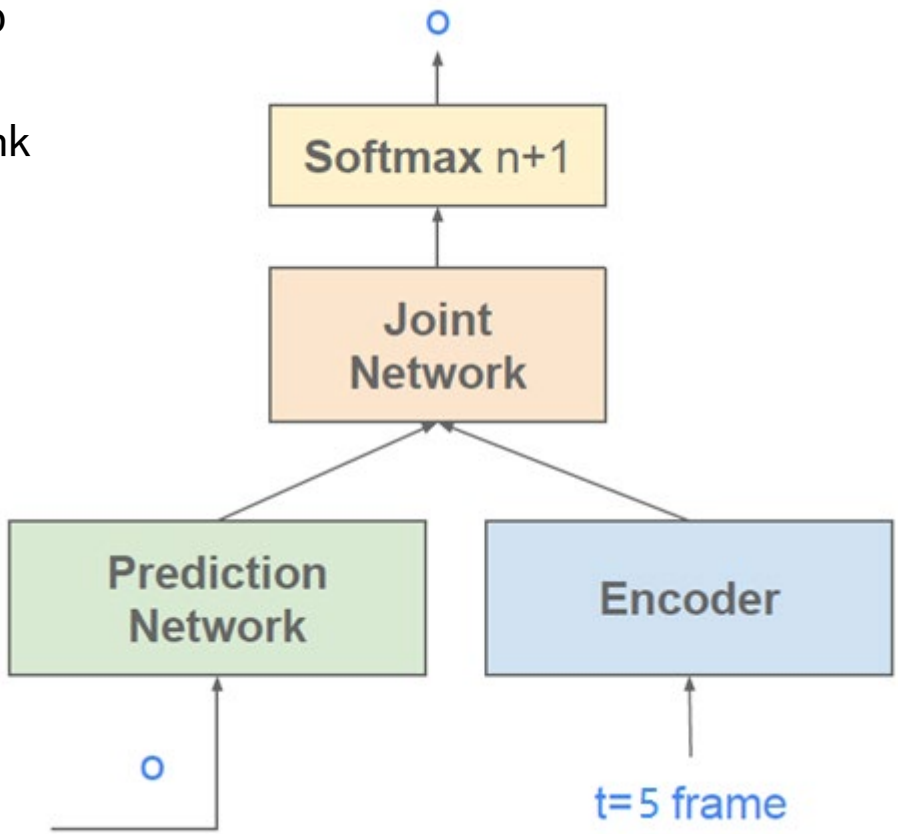


Tara N. Sainath, Towards End-to-End Speech Recognition

RNN-T解码示意：第5步

<sos> <blank> g o o

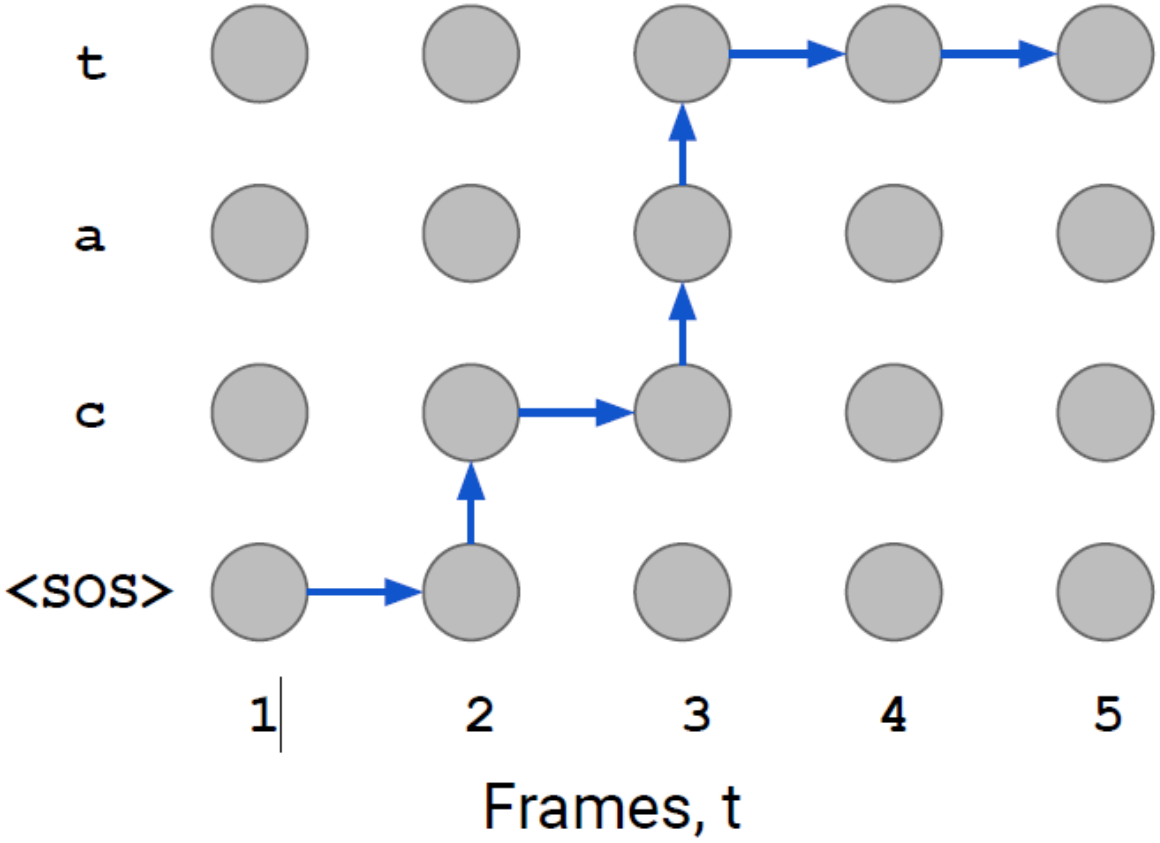
最终删去所有blank
以后生成google。



Tara N. Sainath, Towards End-to-End Speech Recognition

RNN-T的训练

- RNN-T的训练类似与CTC，只是概率路径图有区别。
- 都使用前向后向算法进行计算。



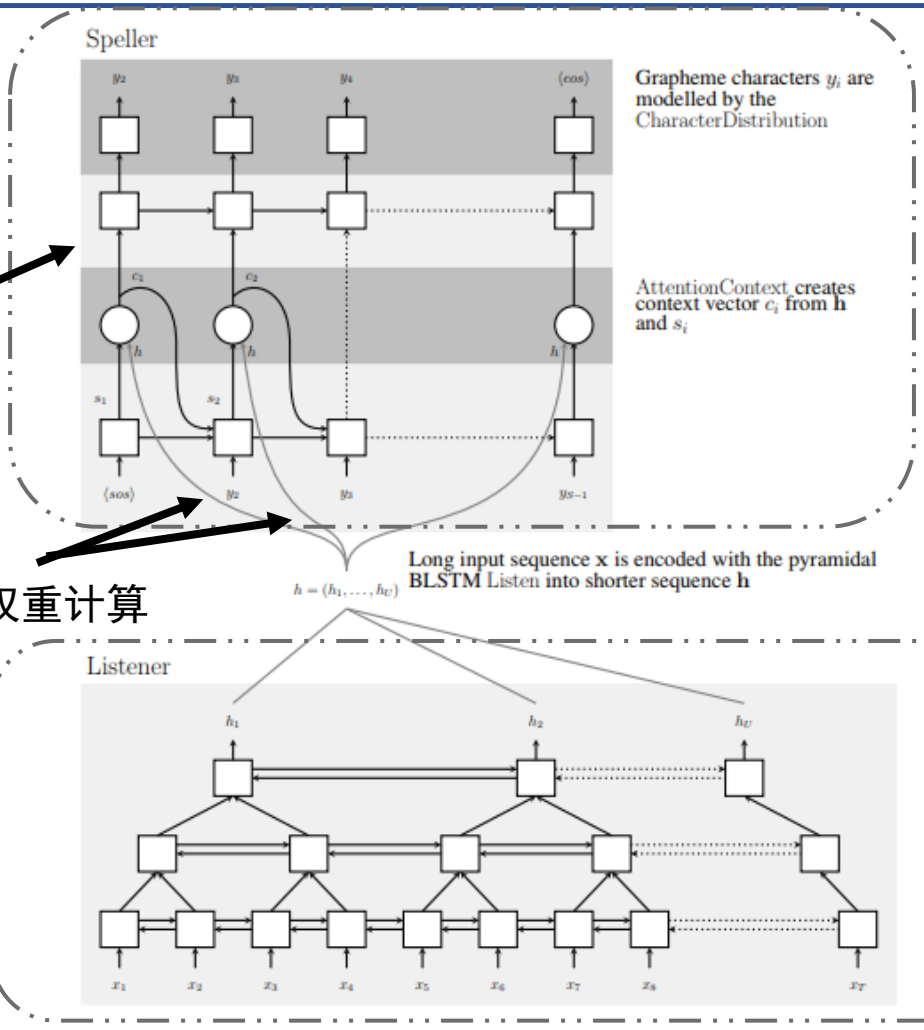
RNN-T的特点

- 针对CTC的不足，RNN-T引入了一个预测网络，类似于语言模型，用来建模语言之间的依赖关系。
- Encoder（编码器）：等同于声学模型
- Prediction Net (预测网络)：等同于语言模型
- Joint Net（联合网络）：结合声学模型状态与语言模型状态就计算输出标记的概率
- 能够建模语言之间的依赖关系，可以流式解码，端到端训练。

提纲

- 解码空间构建
  - WFST回顾
  - 解码图的构建
- 解码搜索
  - 维特比搜索
  - 束搜索
  - 解码实例
- 端到端语音识别模型
  - 为什么要进行端到端语音识别
  - 循环神经网络转换器
  - 基于注意力机制的端到端

基于注意力机制的端到端模型



解码器

注意力权重计算

编码器

编码器：负责将语音特征表征为一组编码向量H

注意力机制：可以看做一种查询机制，其根据当前解码状态去与编码向量计算相关性权重，然后根据权重对编码向量进行加权求和得到上下文向量c

解码器：根据上一时刻的预测结果和上下文向量c计算当前解码结果

上下文向量

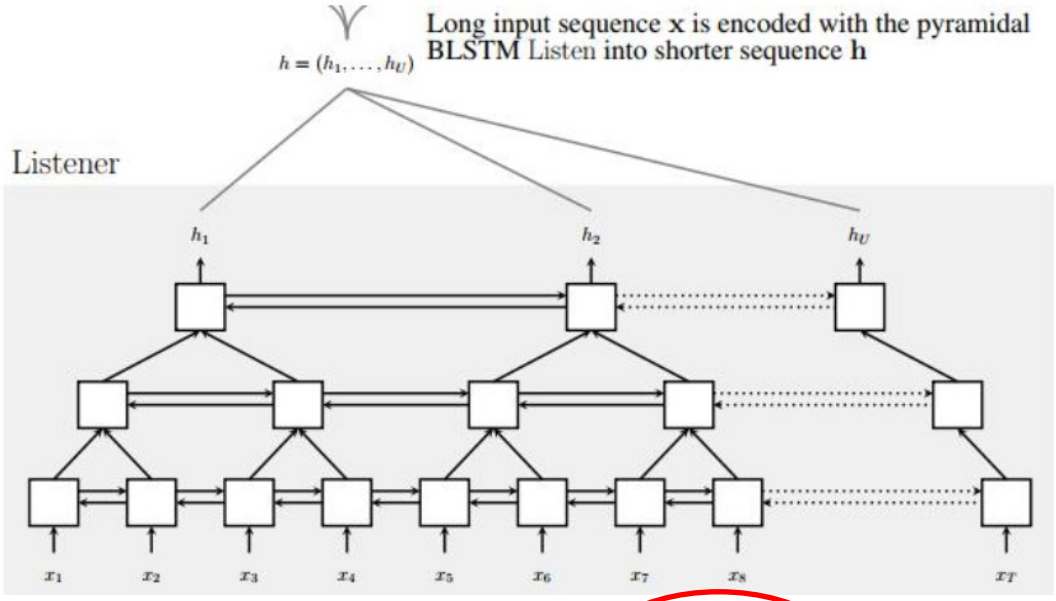
解码器状态

$$c_i = \text{AttentionContext}(s_i, \mathbf{h})$$

编码器输出

金字塔结构的双向LSTM编码器

生成编码后的高层特征序列

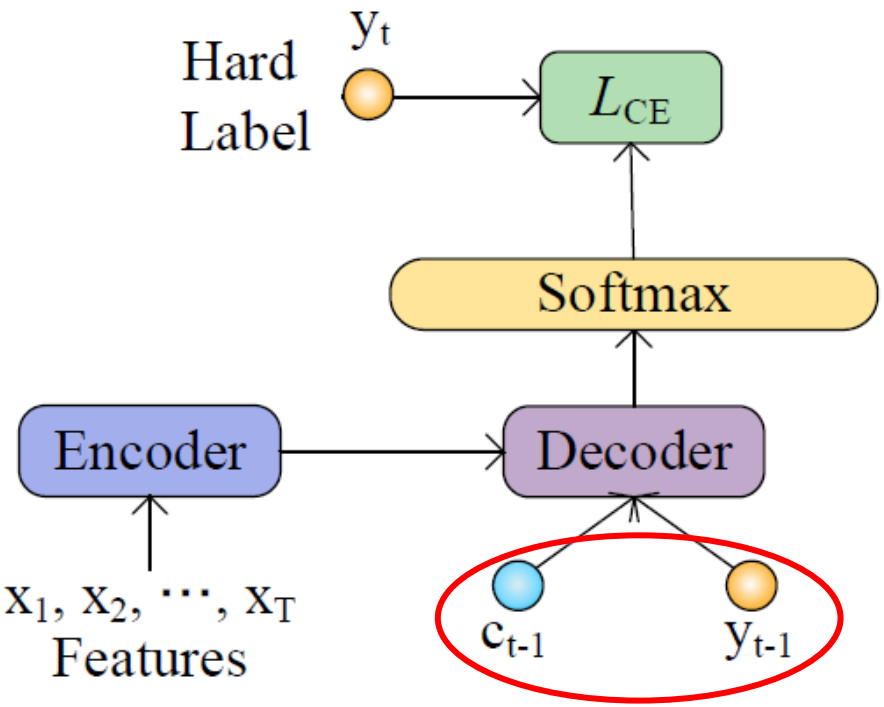


$$h_i^j = \text{pBLSTM}(h_{i-1}^j, \left[h_{2i}^{j-1}, h_{2i+1}^{j-1}\right])$$

实现了特征的抽象以及逐层降采样。

拼接前一层网络的相邻两个状态

基于注意力机制的序列到序列模型



根据上下文向量和上一个字符，
预测下一个字符。

总结

- 本节课介绍了基于加权有限状态转换器的解码图构建方法，然后介绍了解码搜索算法。最后介绍了前沿的端到端语音识别模型。
- 加权有限状态转换器可以融合多种知识，通过加权有限状态转换器，将隐马尔可夫模型，上下文相关音素，发音词典，和语言模型综合起来，构建统一的HCLG解码图。
- 解码就是根据声学分数，和语言模型分数，在HCLG上搜索出最优词序列。本节课着重介绍了维特比搜索，束搜索。
- 端到端语音识别系统利用神经网络同时对语音语言进行建模，联合优化，具有系统体积小，性能好的特点，是目前语音识别发展的前沿。

谢谢！