深蓝学院
shenlanxueyuan.com

语音算法：前沿与应用

语音识别应用系统的搭建

Xiangang Li, Guoguo Chen

搭建语音识别系统

深蓝学院 shenlanxueyuan.com

1 引言

2 语音识别系统与解码器

- 2.1 动态解码器
- 2.2 WFST解码器
- 2.3 Seq2seq解码过程
- 2.4 几种端到端语音识别系统对比

3 语音端点检测（Voice Activity Detector，VAD）

4 讨论

5 课后作业

搭建语音识别系统

深蓝学院 shenlanxueyuan.com

1 引言

深蓝学院 shenlanxueyuan.com

- 语音识别应用

1 引言

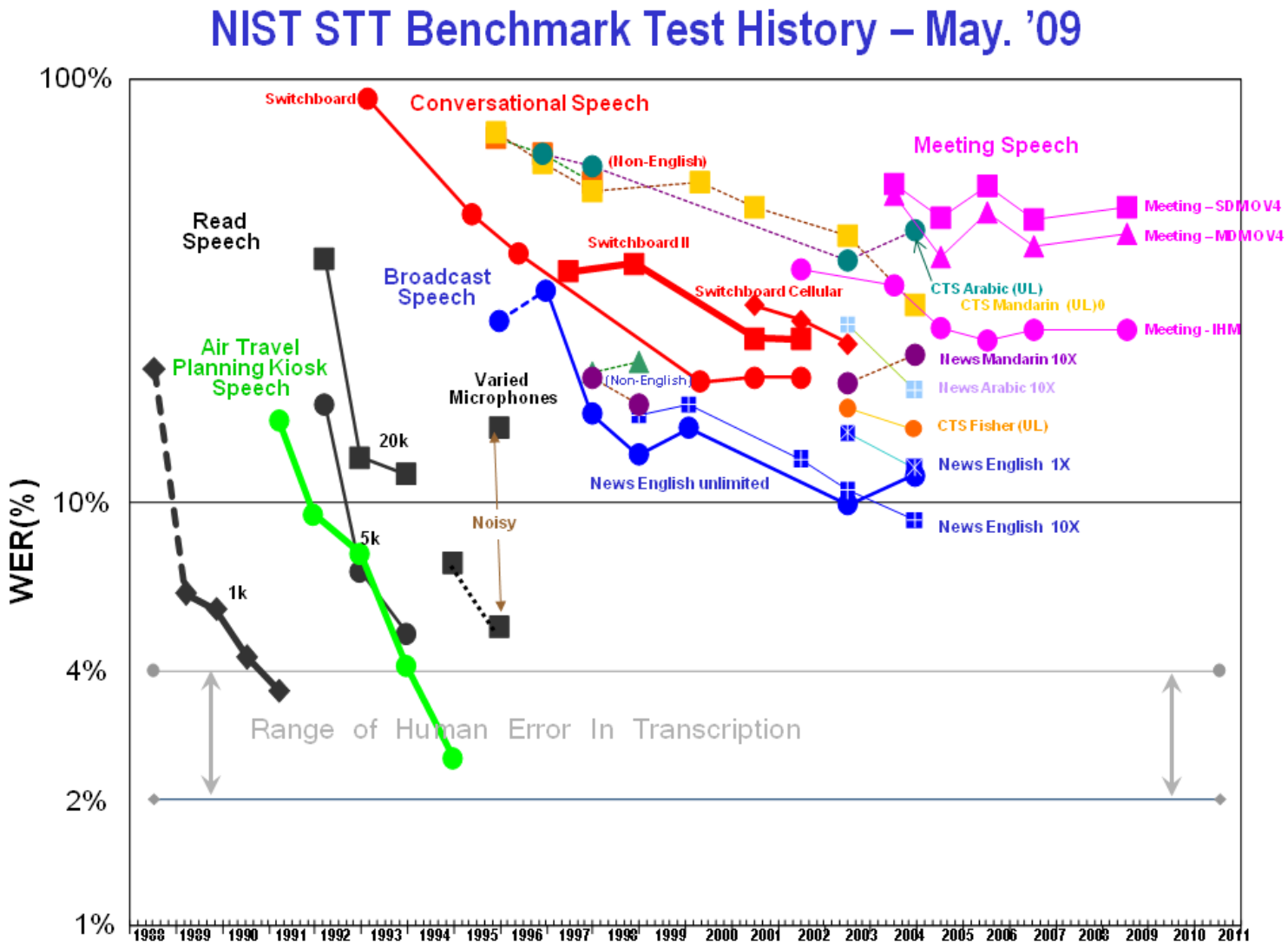
深蓝学院 shenlanxueyuan.com

- 语音识别系统的性能与场景

	近场	远场
人与机器说	语音输入法、语音搜素、语音助手	智能音箱
人与人说	客服、直播	会议、语音监控

- “离开场景谈语音识别性能就是耍流氓”

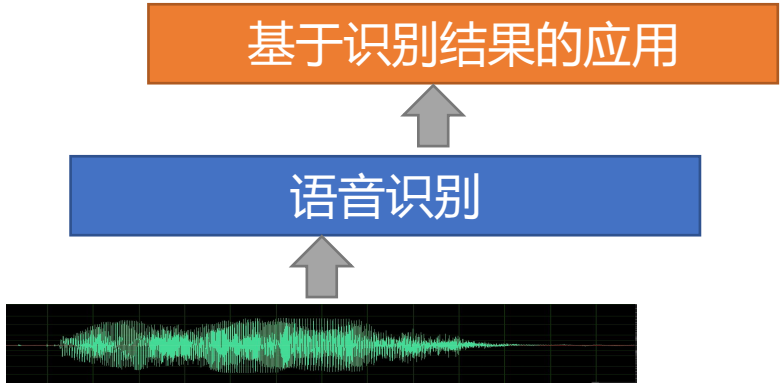
1 引言


NIST STT Benchmark Test History – May. '09
100%
10%
4%
2%
1%
WER(%)
Switchboard
Conversational Speech
(Non-English)
Meeting Speech
Meeting – SDMO V4
Meeting – MDMO V4
Read Speech
Switchboard II
Broadcast Speech
Switchboard Cellular
CTS Arabic (UL)
CTS Mandarin (UL)0
Meeting - IHM
Air Travel Planning Kiosk Speech
(Non-English)
News Mandarin 10X
News Arabic 10X
Varied Microphones
CTS Fisher (UL)
20k
News English 1X
News English unlimited
Noisy
5k
News English 10X
1k
Range of Human Error In Transcription
1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011

1 引言

深蓝学院 shenlanxueyuan.com

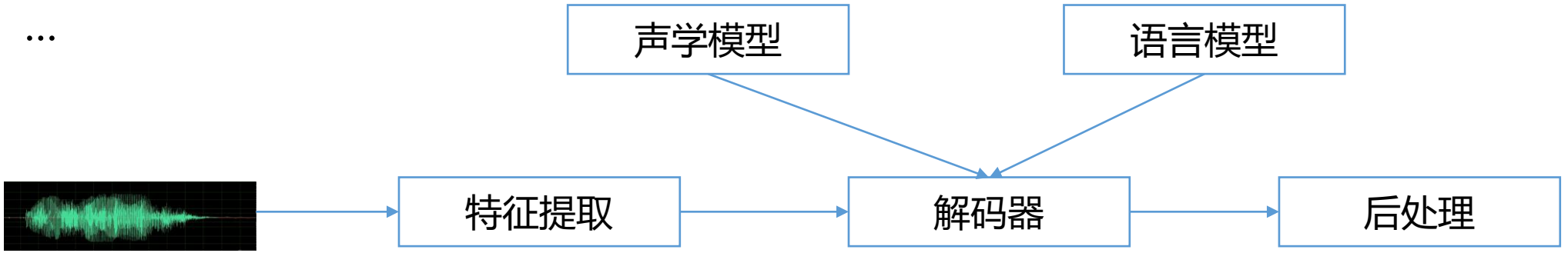
- 语音识别系统
  - Offline整句识别
  - Streaming识别
  - 嵌入式语音识别，云端语音识别



1 引言

深蓝学院 shenlanxueyuan.com

- 语音识别系统
  - 如何实现语音识别解码器：offline/streaming
  - 如何检测语音结束
  - 如何自适应到新的业务场景
  - 后处理
  - …



搭建语音识别系统

深蓝学院 shenlanxueyuan.com

2 语音识别系统与解码器

深蓝学院 shenlanxueyuan.com

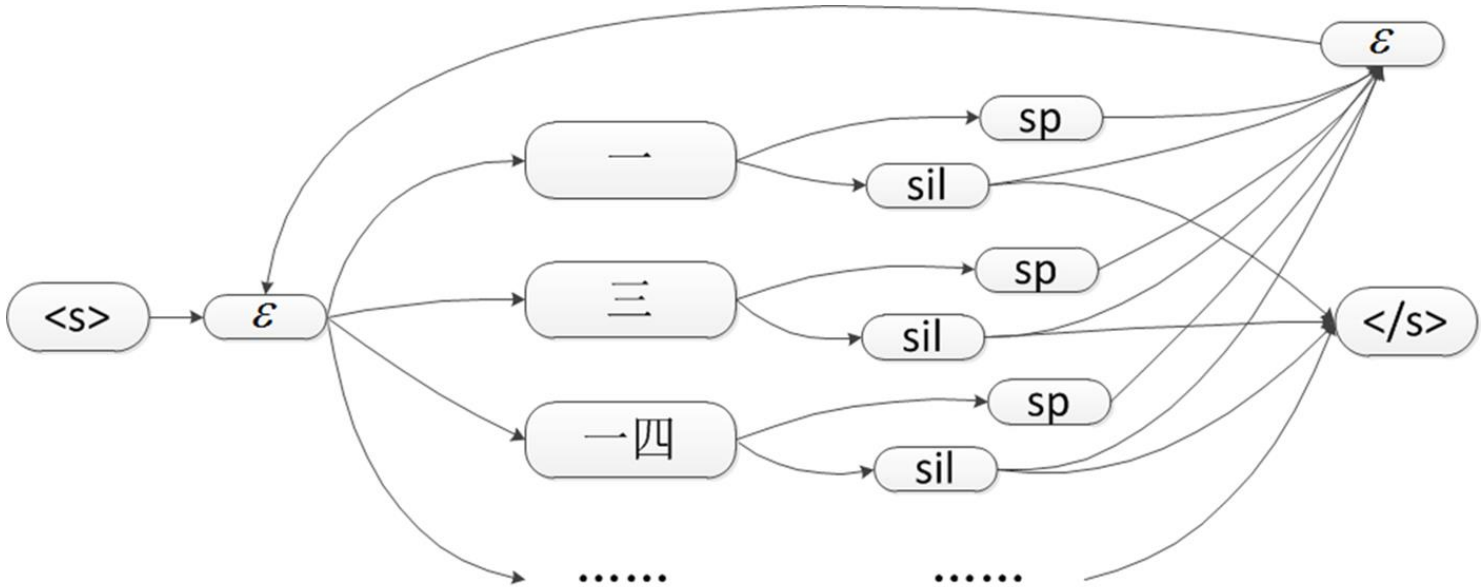
- 解码器做什么？
  - $argmax$
  - 将语音特征转换为文本
  - 输入：语音信号特征、声学模型、语言模型
  - 输出：语音对应的文本结果假设
- 解码器怎么做？
  - 利用输入构建搜索空间（解码网络）
  - 使用搜索算法得到最优结果
- 几个不同时期的解码器
  - HDecode，WFST-decoder，…

2.1 动态解码器——解码器空间的构建

深蓝学院 shenlanxueyuan.com

- 第一步：构建词的解码空间

<s>	sil
</s>	sil
一	y i1
三	s an1
一四	y i1 s i4
…	

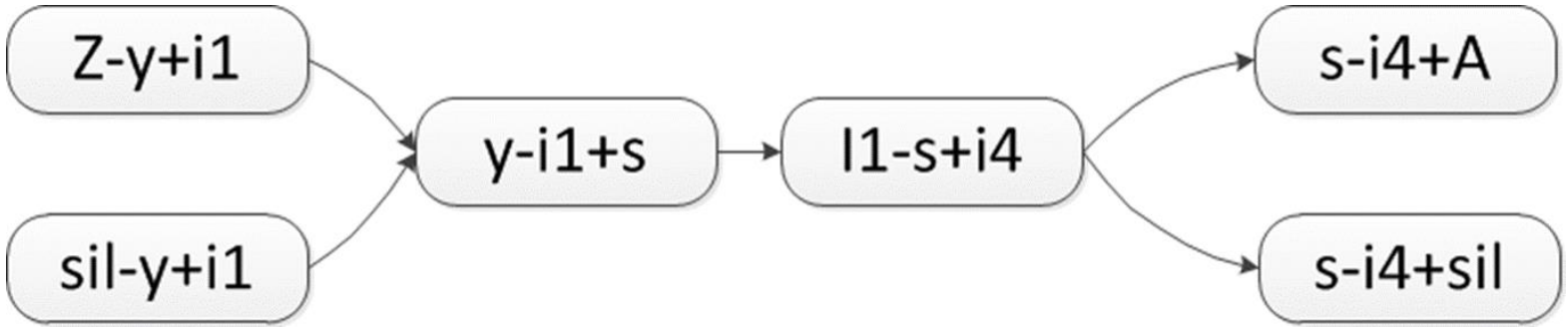


2.1 动态解码器——解码器空间的构建

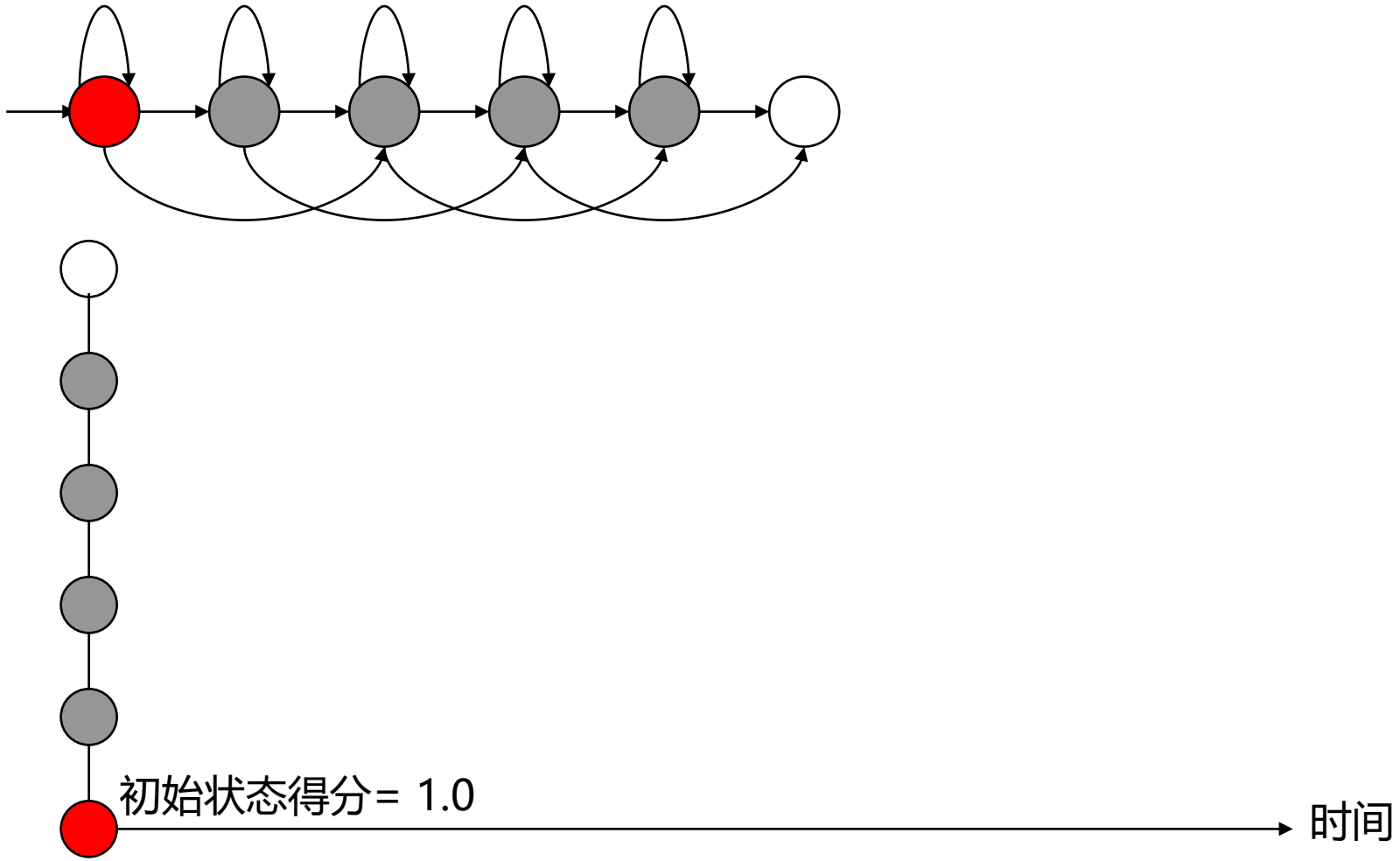
深蓝学院 shenlanxueyuan.com

- 第一步：构建词的解码空间
- 第二步：构建三音子的解码空间

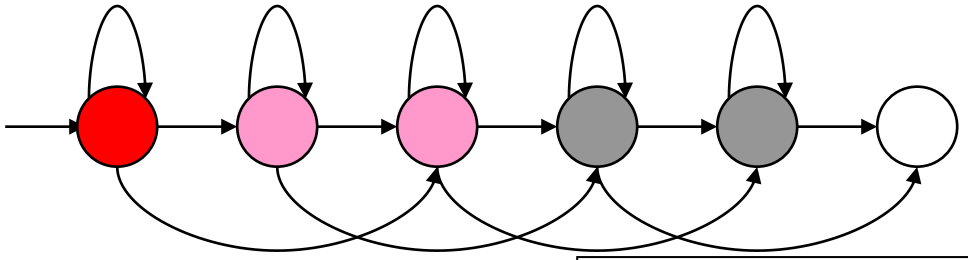
<s>	sil
</s>	sil
一	y i1
三	s an1
一四	y i1 s i4
…	

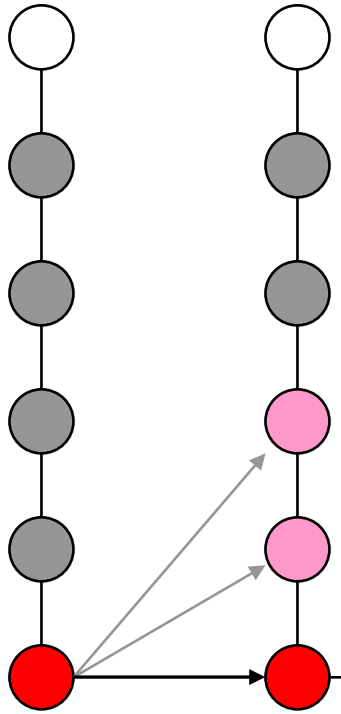


2.1 动态解码器——维特比搜索



2.1 动态解码器——维特比搜索





- 最优路径得分状态
- 得分小于最优路径得分的状态
- 未被扩展的状态

$$P_j(t) = \max_i \left[P_i(t-1) a_{ij} b_j(t) \right]$$

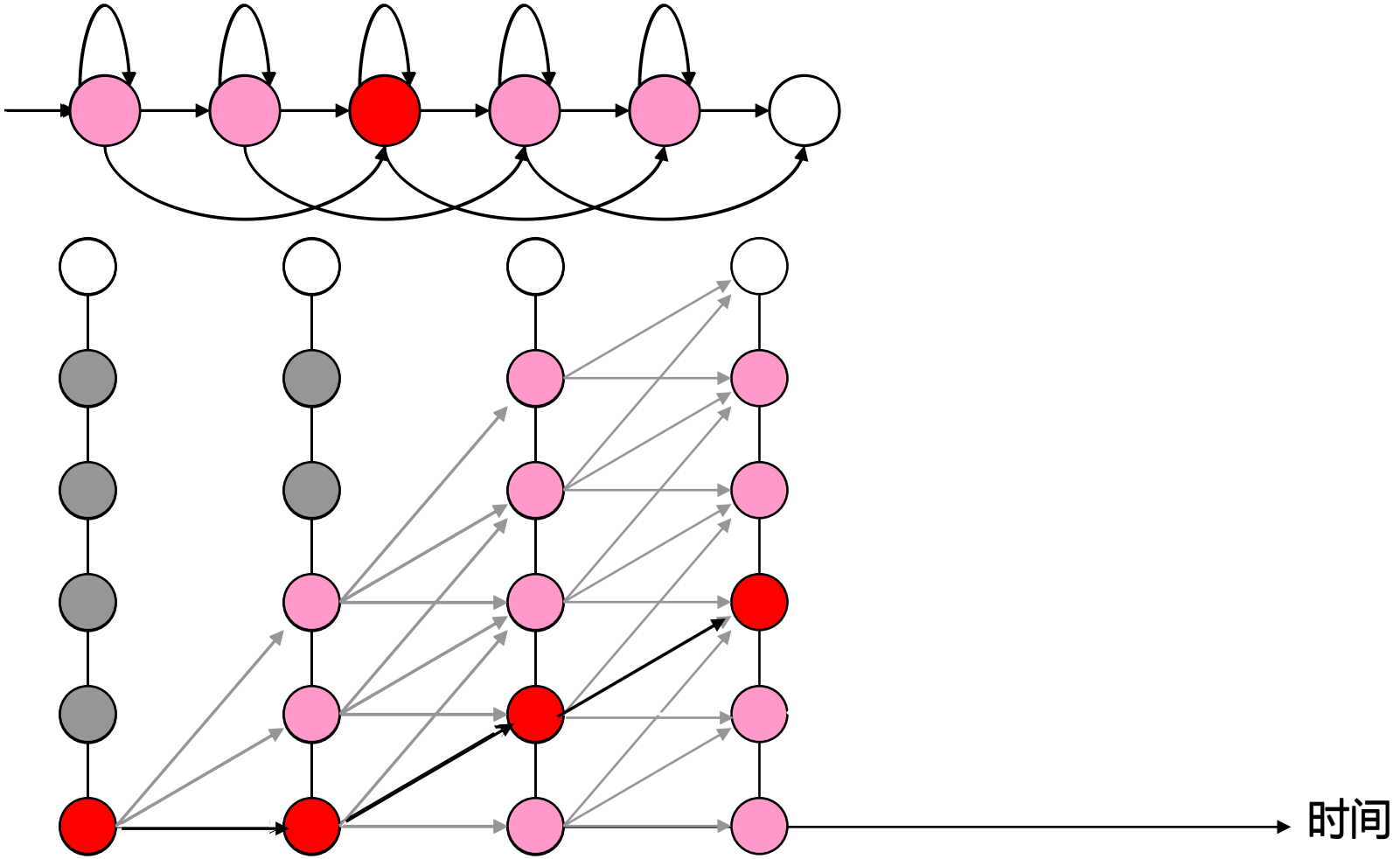
状态跳转概率

t 时刻 j 状态的输出得分

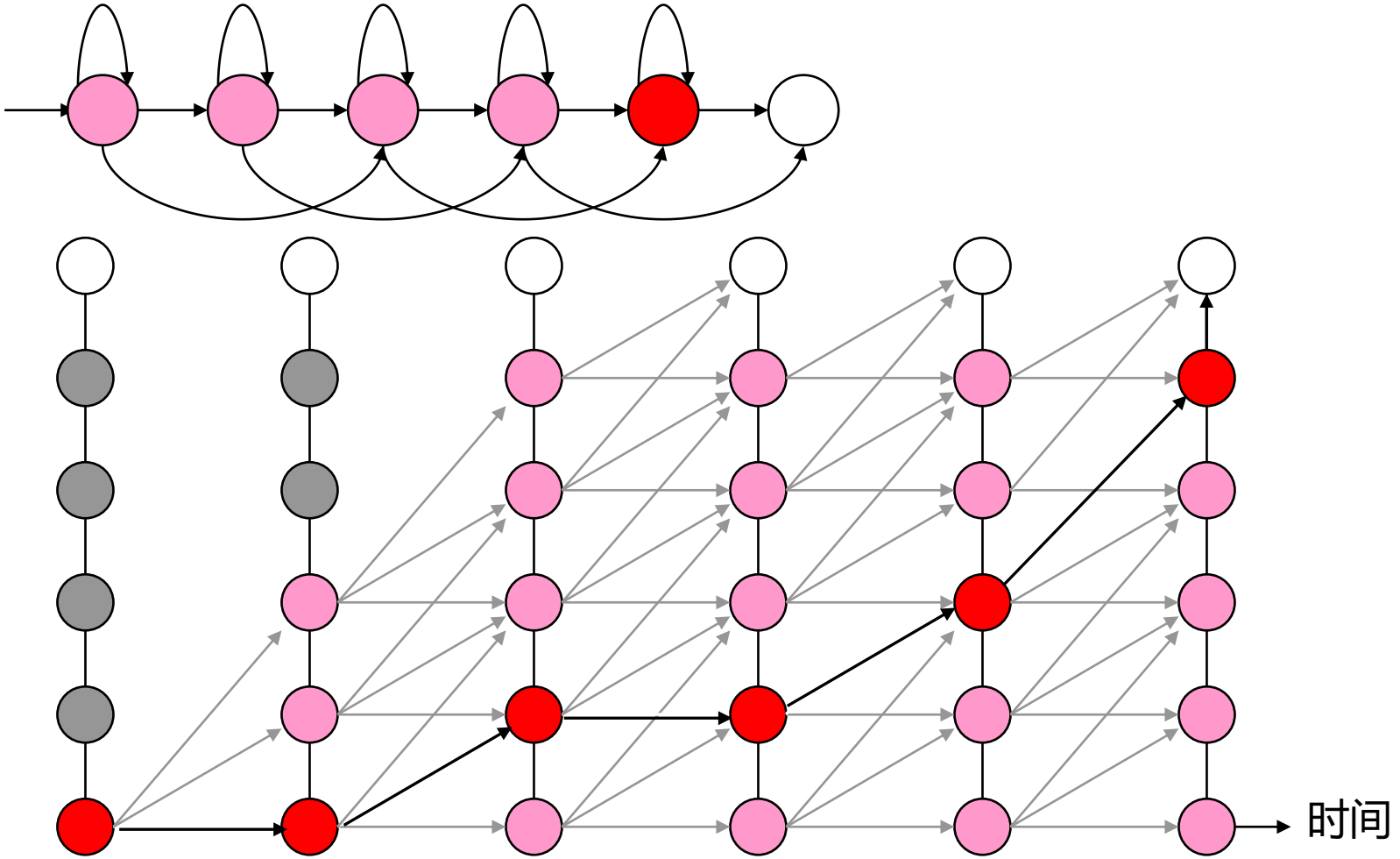
总路径得分，时刻 t，状态 j

时间

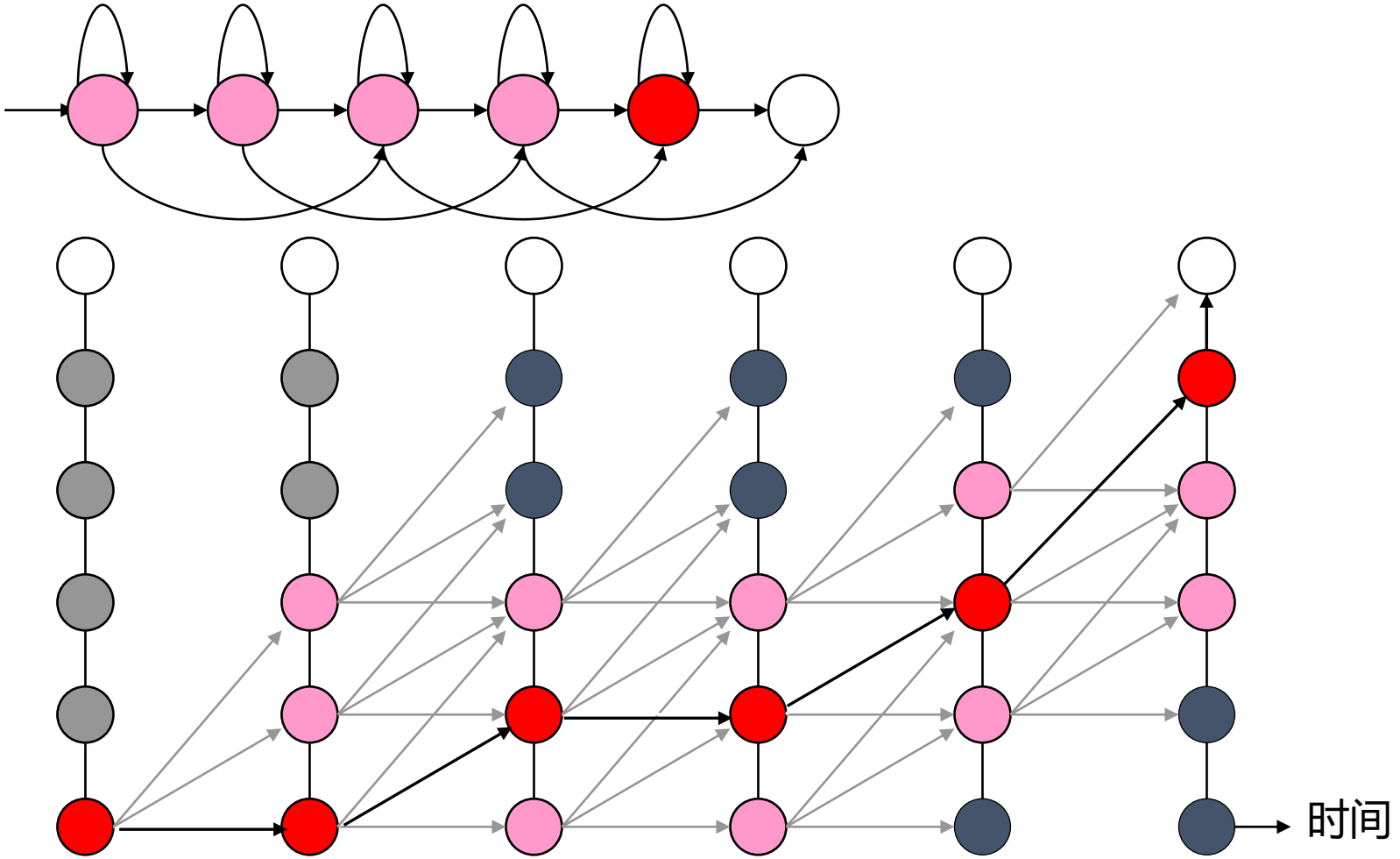
2.1 动态解码器——维特比搜索



2.1 动态解码器——维特比搜索



2.1 动态解码器 – Beam-Viterbi搜索



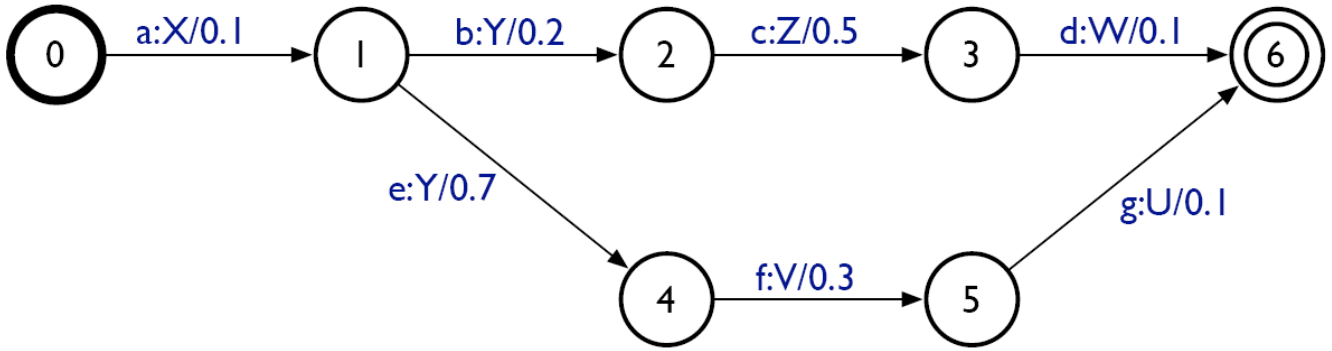
2.1 动态解码器 – Beam-Viterbi搜索

深蓝学院 shenlanxueyuan.com

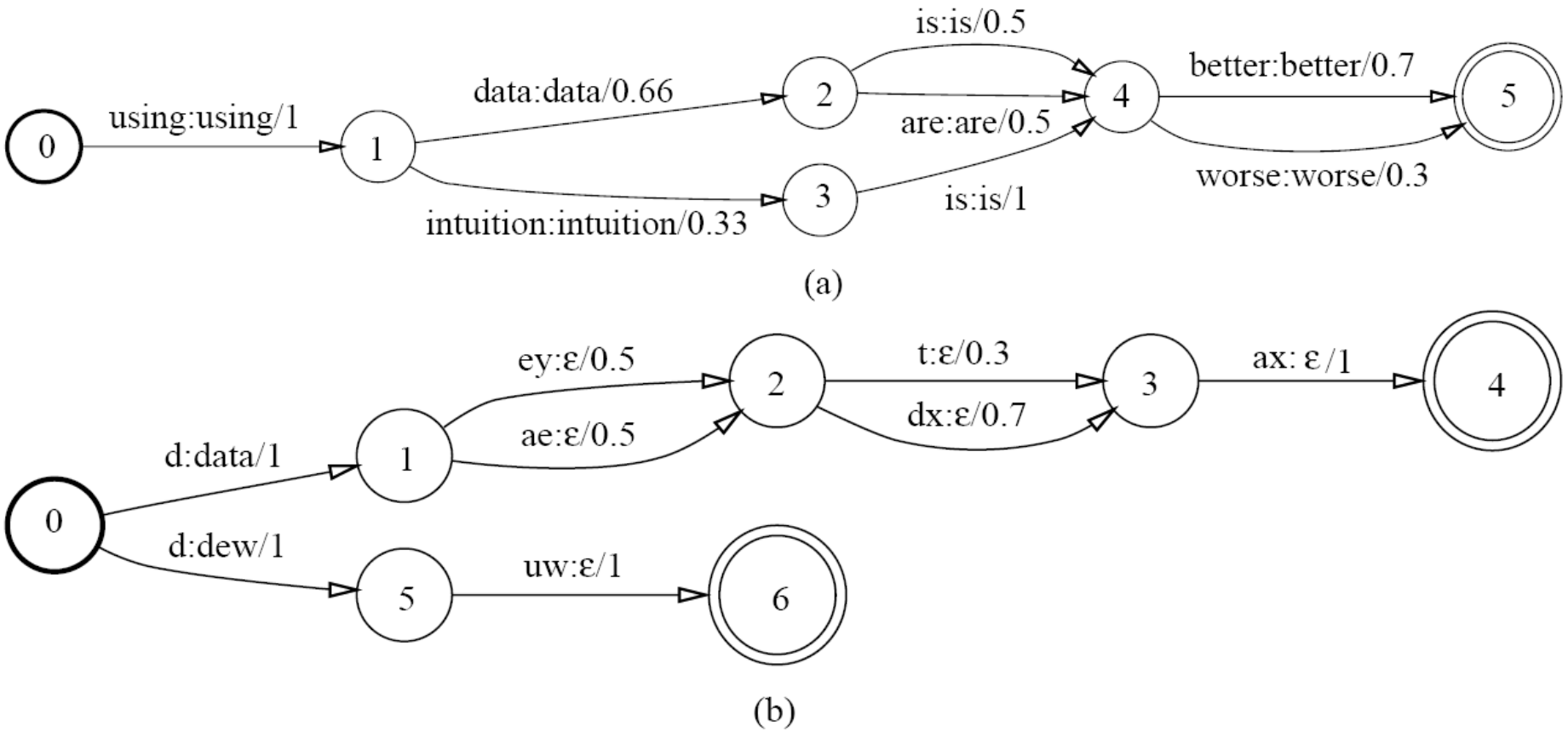
- 与维特比搜索相同点
  - 扩展状态
  - 每个结点保留最高得分
- 与维特比搜索的不同点
  - 每次扩展状态时仅保留得分最高的w个状态
- 语言模型得分前推
  - 语言模型信息越早利用越好

2.2 WFST解码器

- 什么是加权有限状态转录机(Weighted Finite State Transducer , WFST)?
  - 把一个输入串转换成输出串的有限状态自动机
  - 状态间由跳转边连接
  - 每条跳转边有输入标记、输出标记、权重三个元素



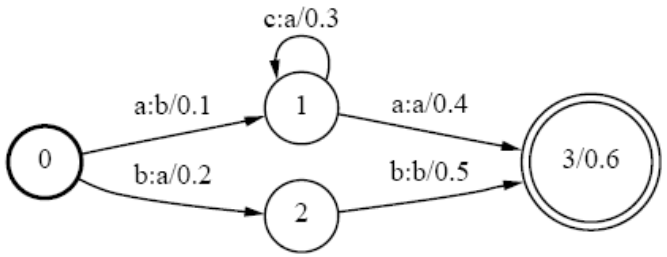
2.2 WFST解码器


0
using:using/1
1
data:data/0.66
2
is:is/0.5
4
are:are/0.5
better:better/0.7
5
worse:worse/0.3
3
is:is/1
intuition:intuition/0.33
(a)
0
d:data/1
1
ey:ε/0.5
ae:ε/0.5
2
t:ε/0.3
dx:ε/0.7
3
ax: ε /1
4
d:dew/1
5
uw:ε/1
6
(b)

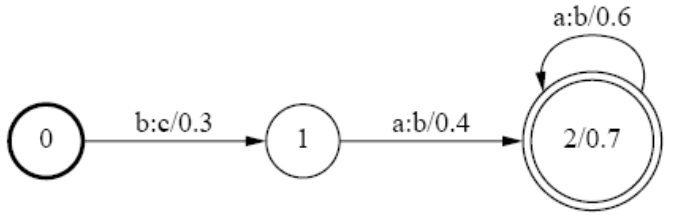
2.2 WFST解码器

- 合并
  - 将多个转换器合并为一个 $A = B \circ C$
- 确定化
  - 对一个输入只给出一个跳转 $\det(A)$
- 最小化
  - 最小化转换器的状态数与边数 $\min(A)$
  - 权重前推

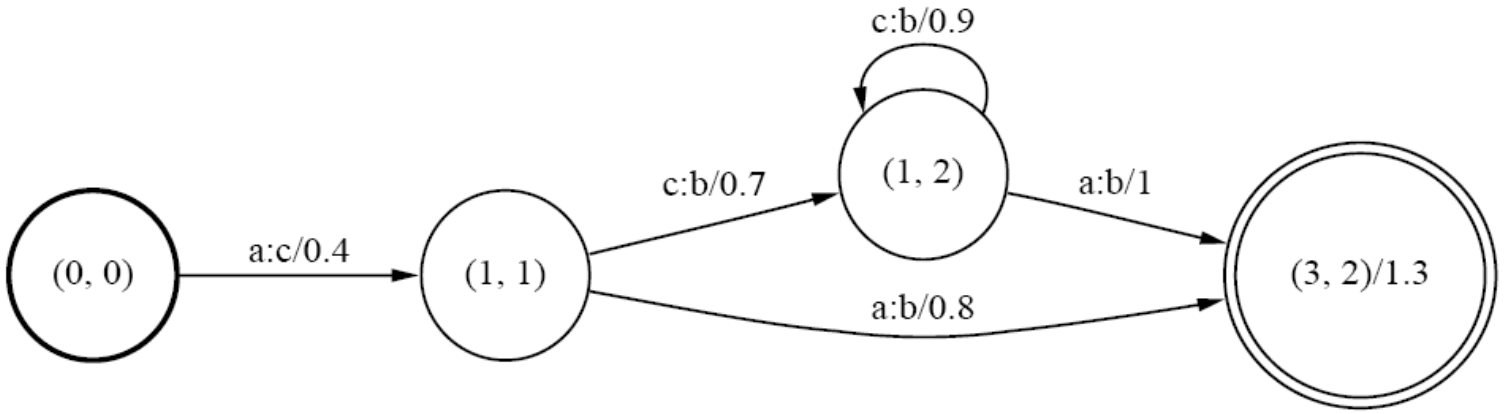
合并


c:a/0.3
a:b/0.1
1
a:a/0.4
0
3/0.6
b:a/0.2
b:b/0.5
2

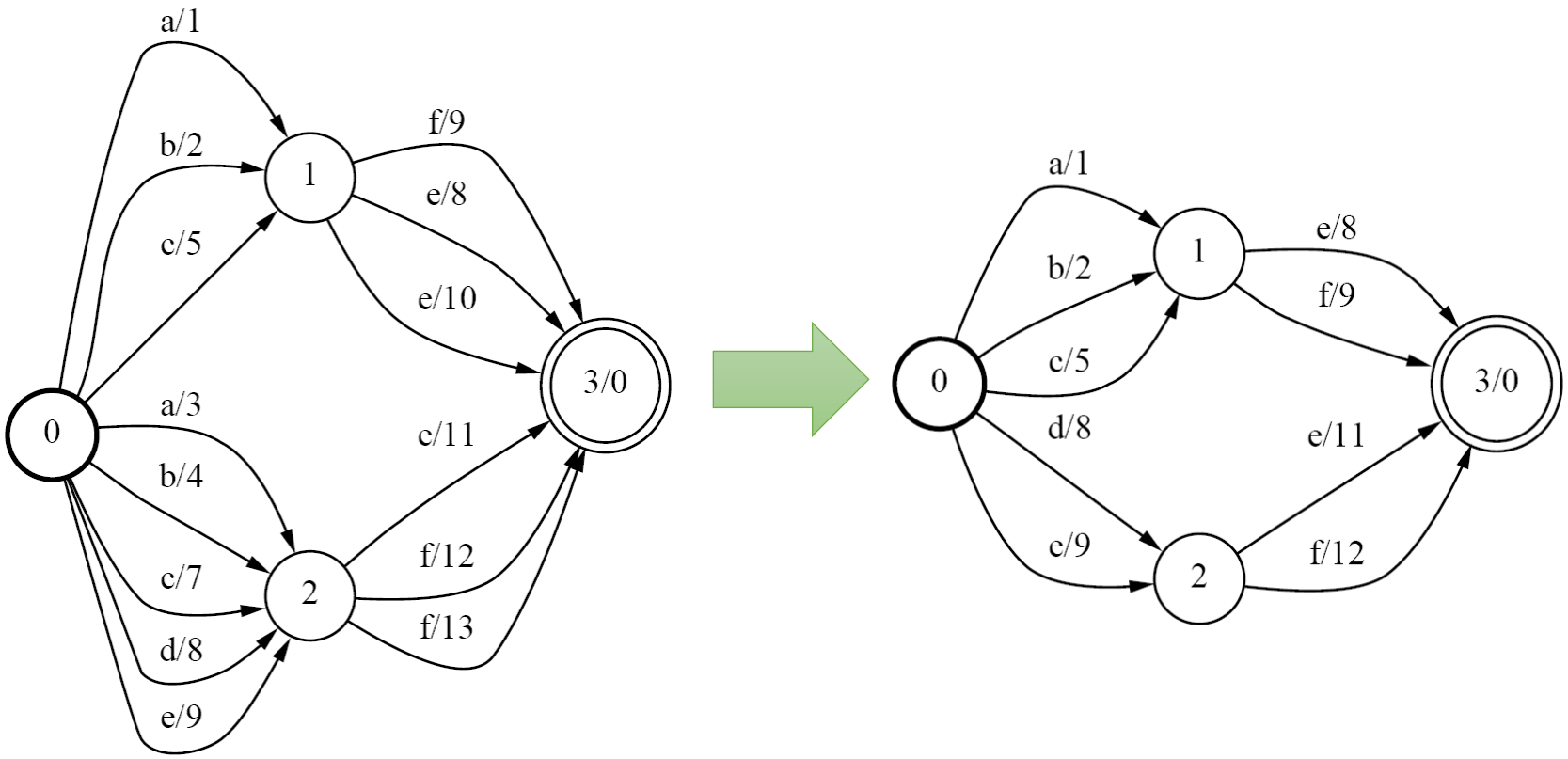
(a)


a:b/0.6
0
b:c/0.3
1
a:b/0.4
2/0.7

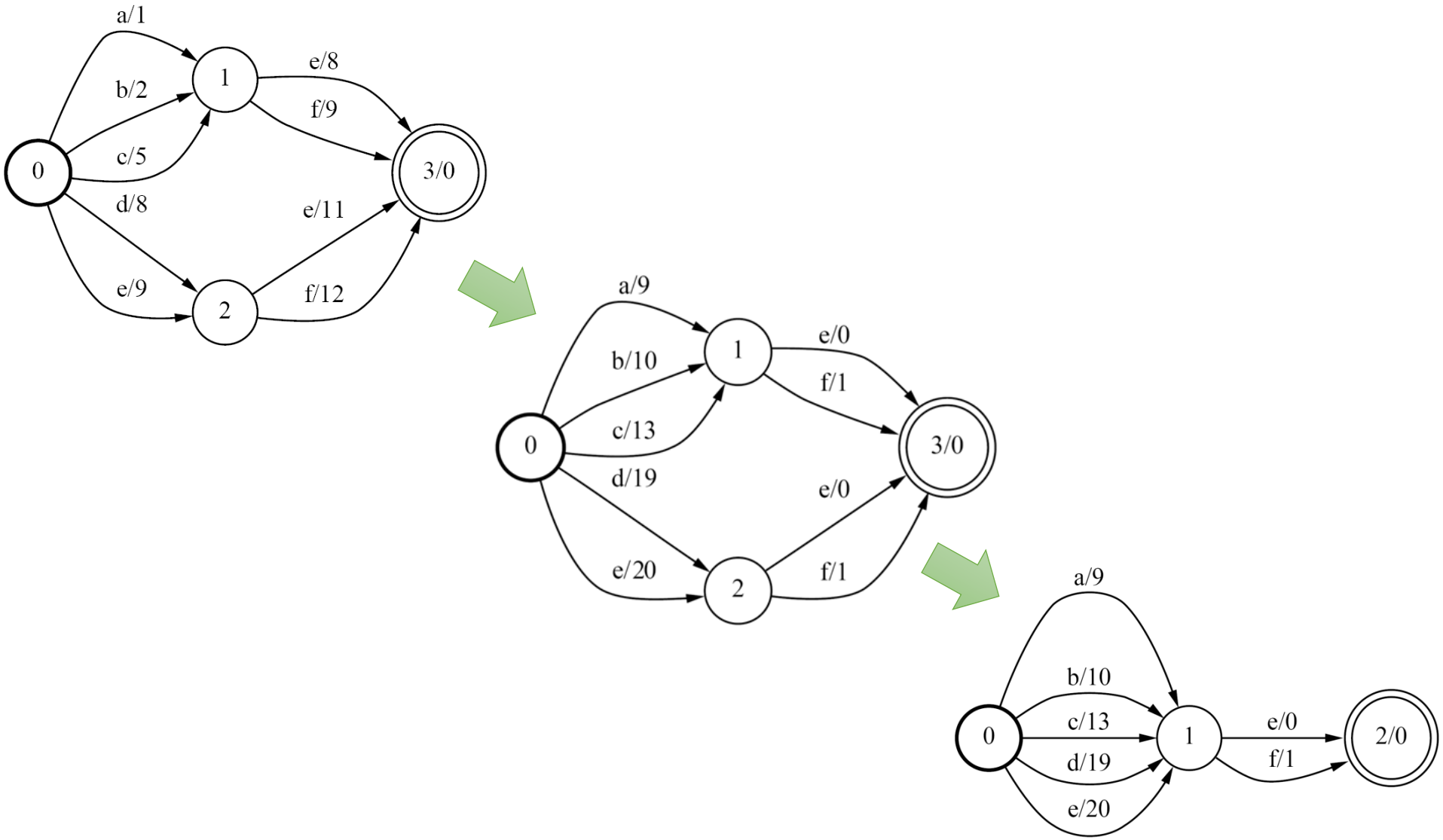
(b)


c:b/0.9
c:b/0.7
(1, 2)
a:b/1
(0, 0)
a:c/0.4
(1, 1)
(3, 2)/1.3
a:b/0.8

确定化


a/1
b/2
c/5
f/9
e/8
e/10
1
0
3/0
a/3
b/4
c/7
d/8
e/9
e/11
f/12
f/13
2
a/1
b/2
c/5
d/8
e/9
e/8
f/9
e/11
f/12
1
0
3/0
2

权重前推与最小化


a/1
b/2
c/5
d/8
e/9
e/8
f/9
e/11
f/12
0
1
2
3/0
a/9
b/10
c/13
d/19
e/20
e/0
f/1
e/0
f/1
0
1
2
3/0
a/9
b/10
c/13
d/19
e/20
e/0
f/1
0
1
2/0

WFST的描述能力

深蓝学院 shenlanxueyuan.com

- WFST可以描述：
  - 文法、发音词典、上下文相关音子、HMM/CTC拓扑、语言模型、…
- WFST是一个自动机
  - 标准操作：
    - 合并、优化、搜索、裁剪、…
  - 扩展操作：
    - 找最优路径、找N-best路径、删除不可达路径和跳转、…

WFST的描述能力

- 多层知识融合

$H \circ C \circ L \circ G$

G：语言模型

L：发音词典

T：上下文相关音子

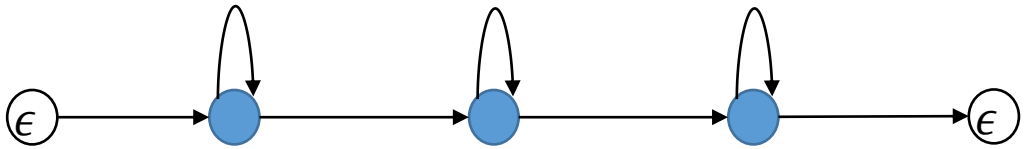
H：HMM

$T \circ L \circ G$

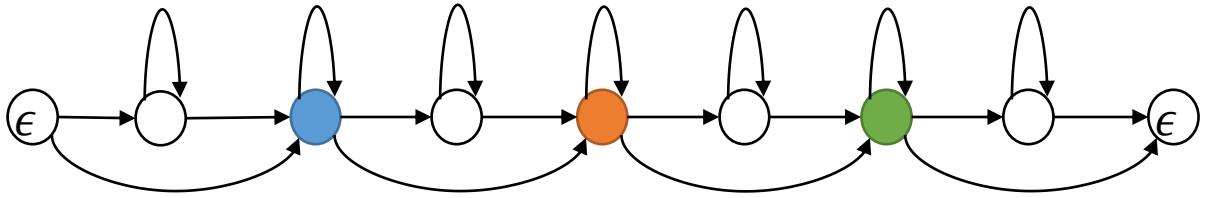
G：语言模型

L：发音词典

T：CTC Token



- 思考题
  - 如何用WFST描述HMM以及CTC



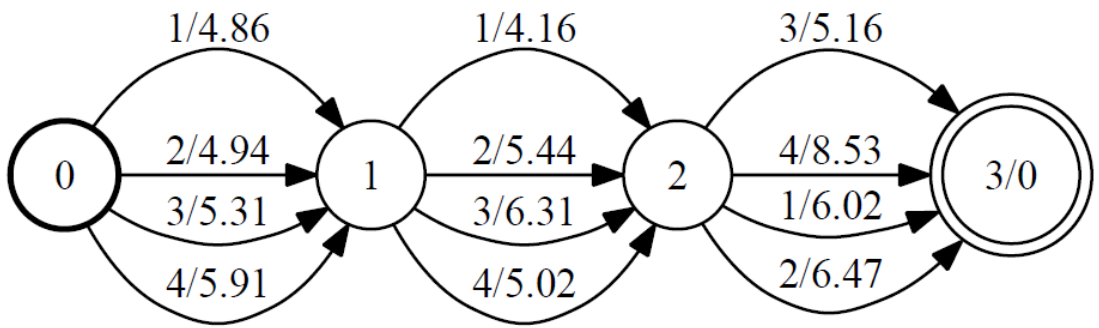
2.2 WFST解码器

深蓝学院 shenlanxueyuan.com

- 解码网络优化

$$HCLG = \min(\det(H \circ C \circ L \circ G))$$

- 声学得分



- 解码过程

$$S \equiv U \circ HCLG$$

$$b = bestpath(S, \alpha)$$

2.2 WFST解码器

深蓝学院 shenlanxueyuan.com

- 思考题1：基于WFST实现语音文字的强制对齐（Force Alignment）
  - 解码网络如何构建：

$$WFST = H \circ C \circ L \circ G_{sentence}$$

  - 如何基于一句话构建其WFST $G_{sentence}$
  - 标准的CTC模型为什么不适合做对齐

- 扩展思考题2：基于WFST实现基于N-Gram语言模型的序列标注/转换任务
  - 拼音输入法 $WFST = L \circ G$
  - 分词、注音、加标点
  - …

2.2 WFST解码器

深蓝学院 shenlanxueyuan.com

- WFST面临的问题：大语言模型

$HCLG_{\text{small}} = H \circ C \circ L \circ G_{\text{small}}$

$S \equiv U \circ HCLG_{\text{small}} \circ G_{\text{big-small}}$

G_{small}：小语言模型

$G_{\text{big-small}}$：大小语言模型差值

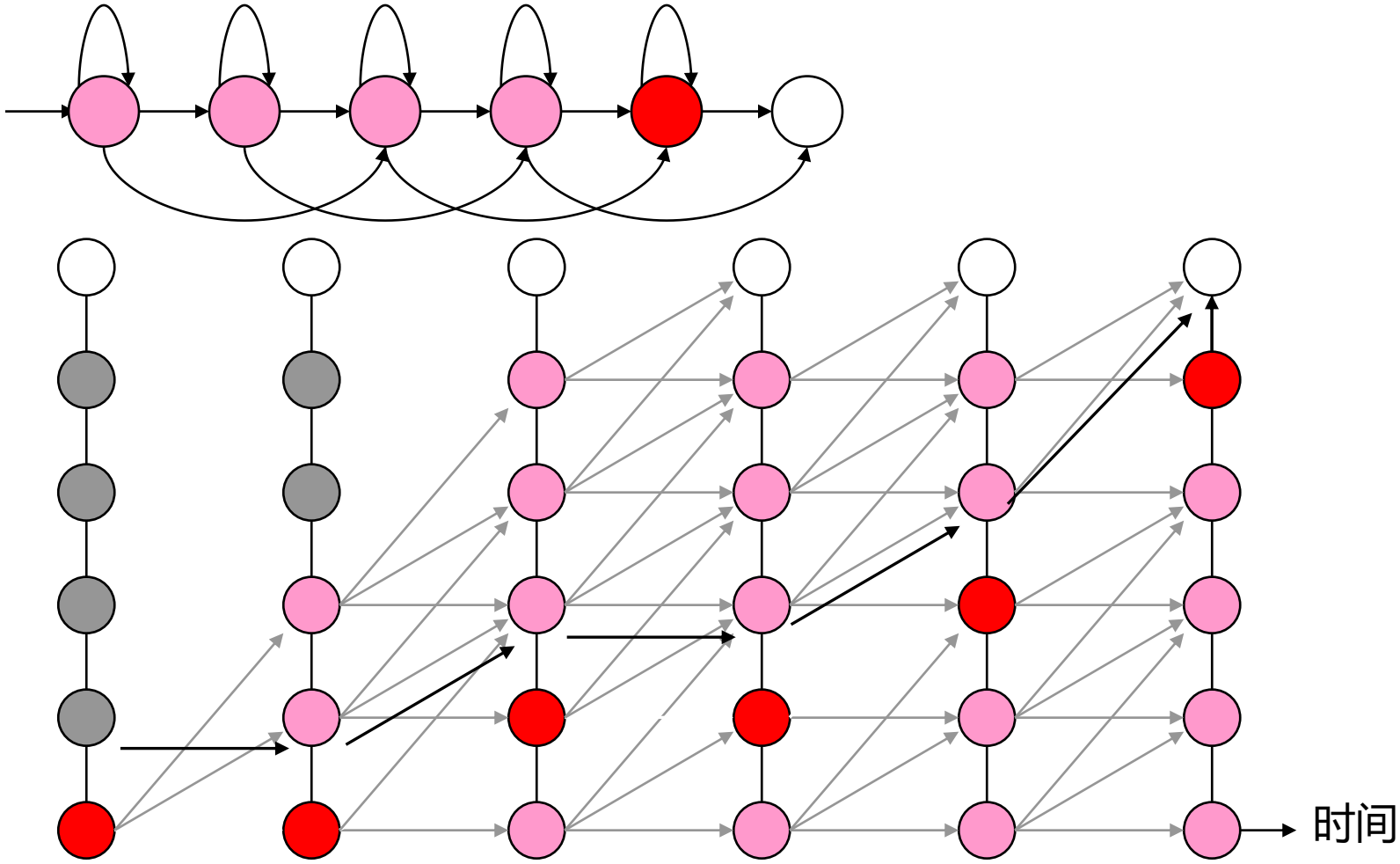
```
fst::BackoffDeterministicOnDemandFst<StdArc> old_lm_dfst(*old_lm_fst);
fst::BackoffDeterministicOnDemandFst<StdArc> new_lm_dfst(*new_lm_fst);
fst::ComposeDeterministicOnDemandFst<StdArc> compose_dfst(&old_lm_dfst,
                                                          &new_lm_dfst);
fst::CacheDeterministicOnDemandFst<StdArc> cache_dfst(&compose_dfst);

BiglmFasterDecoder decoder(*decode_fst, decoder_opts, &cache_dfst);

DecodableAmDiagGmmScaled gmm_decodable(am_gmm, trans_model, features,
                                       acoustic_scale);
decoder.Decode(&gmm_decodable);
```

2.2 WFST解码器

深蓝学院 shenlanxueyuan.com

- Streaming
  - 每个时刻输出当前得分最高的路径对应的识别结果

$$b_t = bestpath(S, \alpha, t)$$
$$r_t = traceback\ (b_t)$$



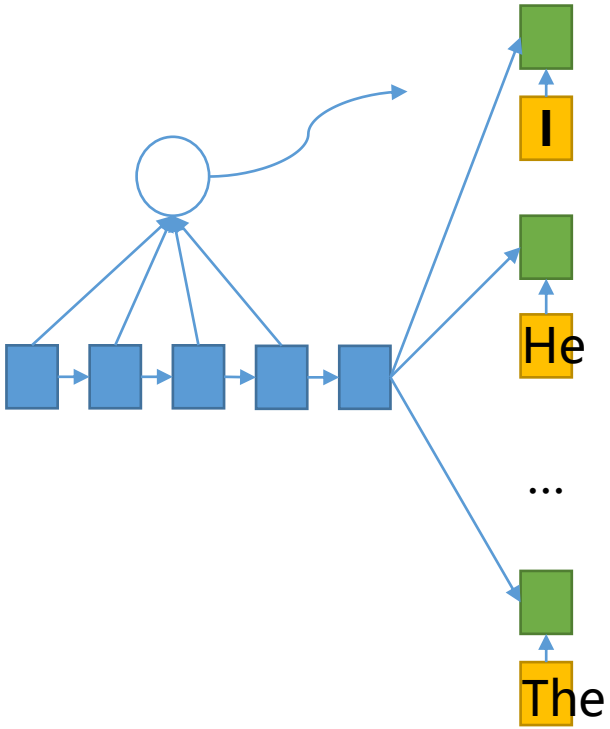
2.3 Seq2seq解码过程

深蓝学院 shenlanxueyuan.com

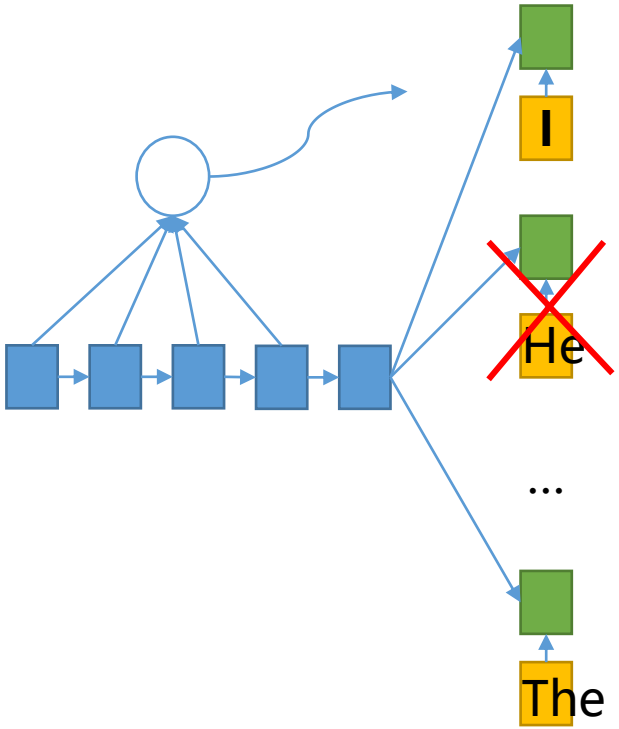
- 与HMM/CTC解码过程的差别
  - 帧同步 & 输出同步
- Seq2seq解码过程
  - 在解码的每一步，保存k个最有可能的结果
  - 当相关路径输出<end>时，代表路径结束

```
def argmax_decode(inputs):
  encoder_output, decoder_hidden = encoder(inputs)
  decoder_input = tf.expand_dims([target_lang.word_index['<START>']], 0)
  for t in range(max_length):
    predictions, decoder_hidden, attention_weights = decoder(decoder_input,
                                                             decoder_hidden,
                                                             encoder_output)

    predicted_idx = tf.argmax(predictions[0])
    results += target_lang.index_word[predicted_idx] + ' '

    if target_lang.index_word[predicted_idx] == '<END>':
      return results

    decoder_input = tf.expand_dims([predicted_idx], 0)

  return results
```

2.3 Seq2seq解码过程

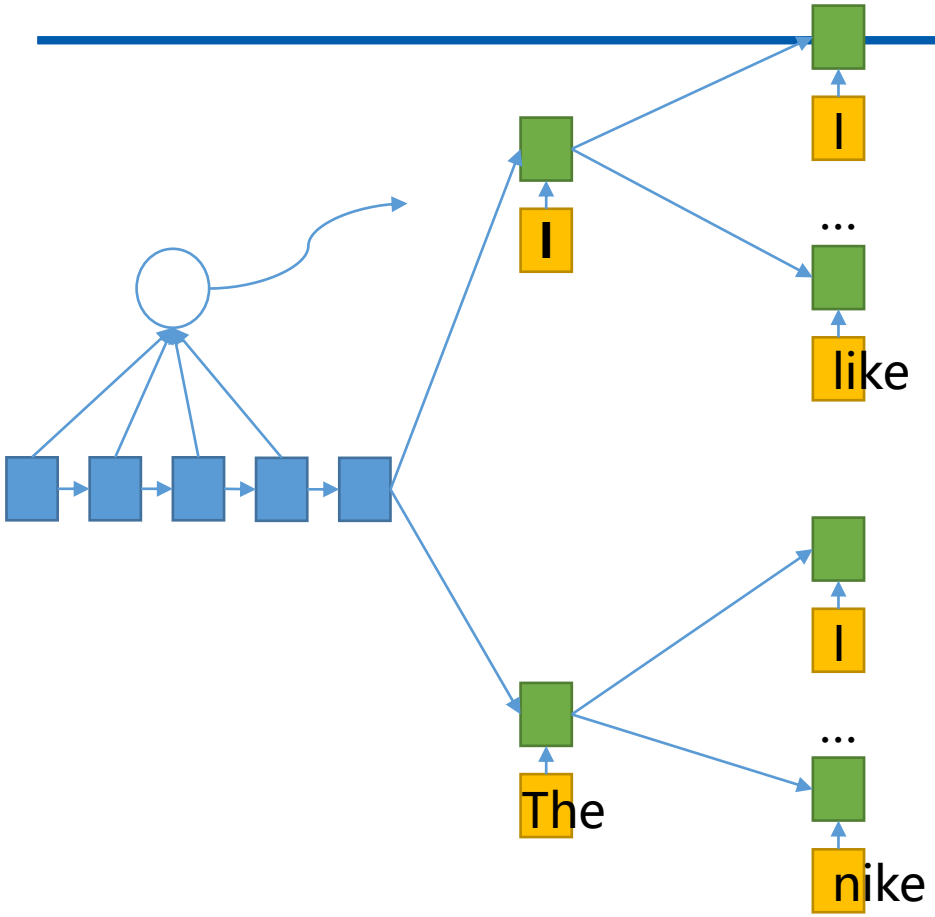


2.3 Seq2seq解码过程



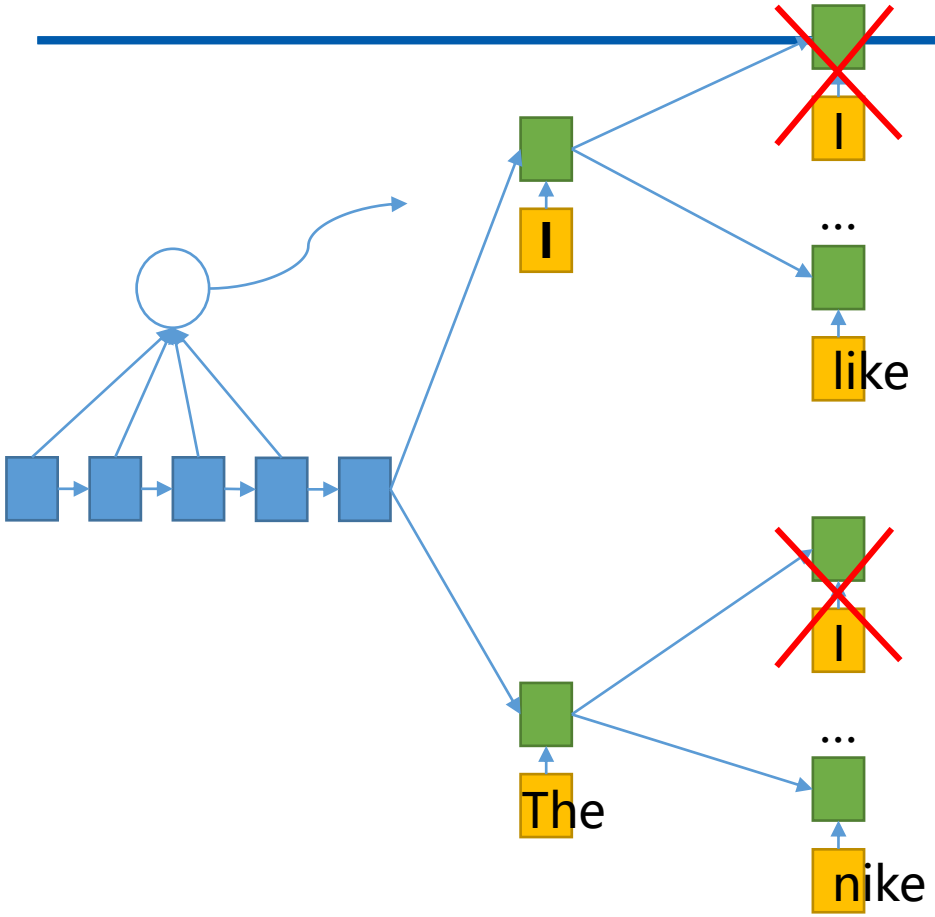
- 每一个时刻，只保留TopK

2.3 Seq2seq解码过程



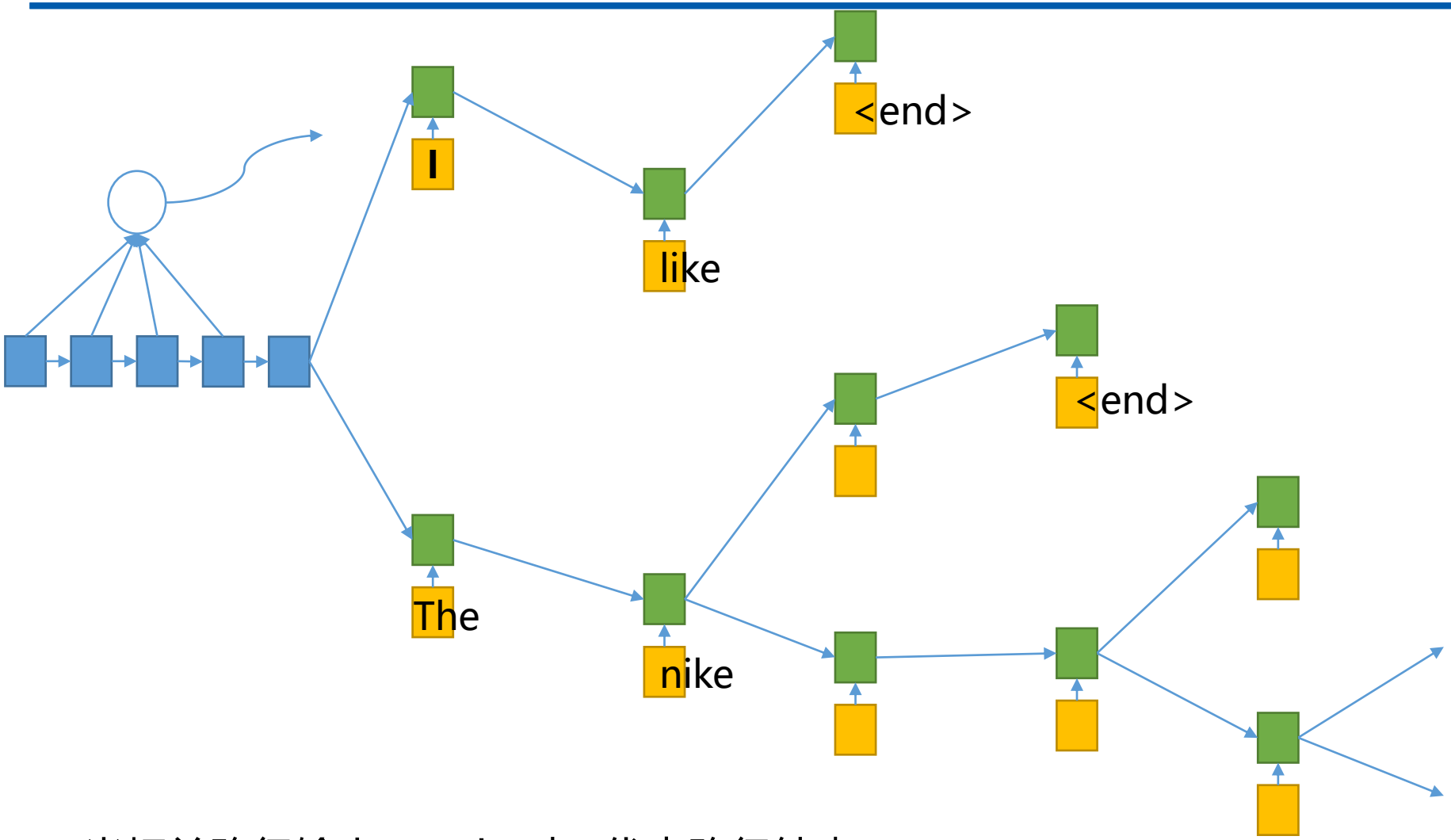
- 每一个时刻，只保留TopK

2.3 Seq2seq解码过程



- 每一个时刻，只保留TopK

2.3 Seq2seq解码过程

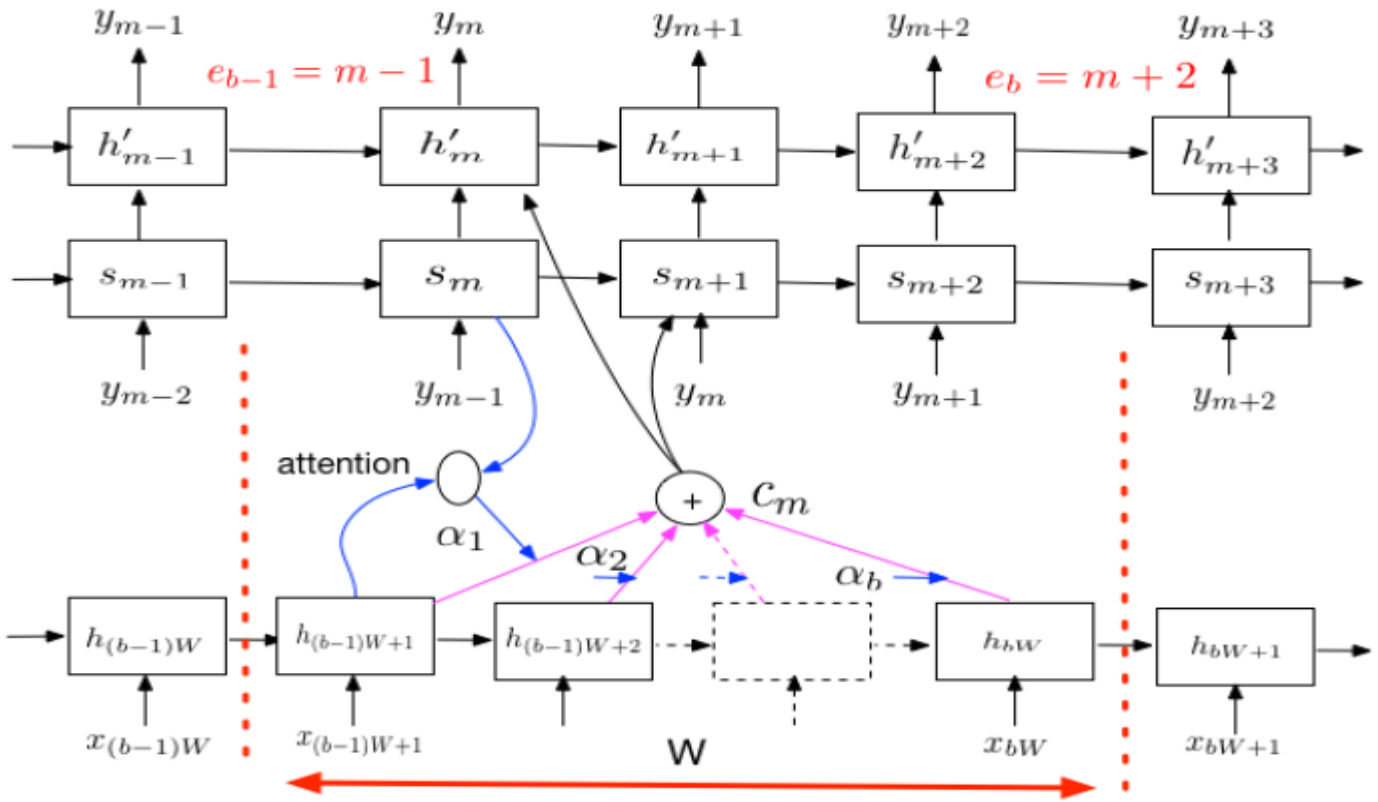


- 当相关路径输出<end>时，代表路径结束

Streaming Seq2seq语音识别

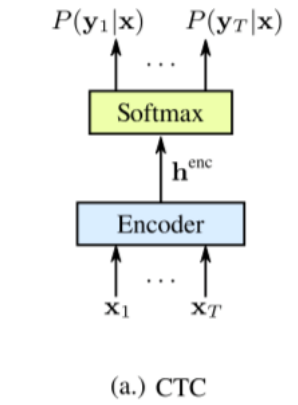
深蓝学院 shenlanxueyuan.com

- 难点
  - 最初版本的Seq2seq基于完整输入序列计算Attention Weights

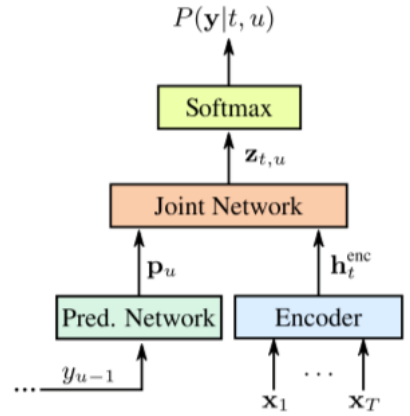


2.4 几种端到端语音识别系统对比

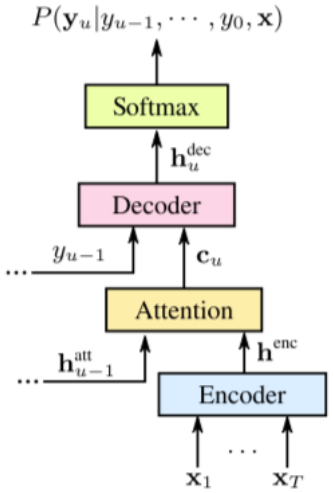
1. Alex Graves. Sequence Transduction with Recurrent Neural Networks. ICASSP 2012.
2. E. Battenberg, J. Chen, R. Child, and et.al. Exlploring Neural Transducers for End-to-End Speech Recognition. arXiv:1707.0741 2017
3. R. Prabhavalkar, K. Rao, T. Sainath, B. Li, L. Johnson, and N. Jaitly. A Comparison of Sequence-to-Sequence Models for Speech Recognition. Interspeech 2017
4. K. Rao, H. Sak, R. Prabhavalkar. Exploreing Architectures, Data and Units for Streaming End-to-End Speech Recognition with RNN-Transducer. arXiv: 1801.00841. 2018
5. Y. He, T. Sainath, R. Prabhavalkar and et al. Streaming End-to-end Speech Recognition For Mobile Devices. arXiv: 1811.06621. 2018
6. Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermatt, S. Koo, S. Kuma Transformer Transducer: a Streamable Speech Recognition Mode with Transformer Encoders and RNN-T Loss



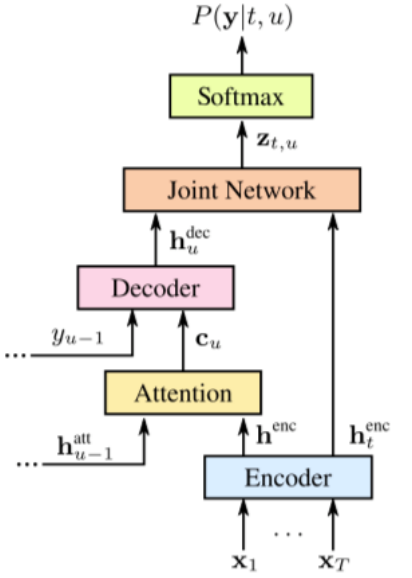
(a.) CTC



(b.) RNN-Transducer



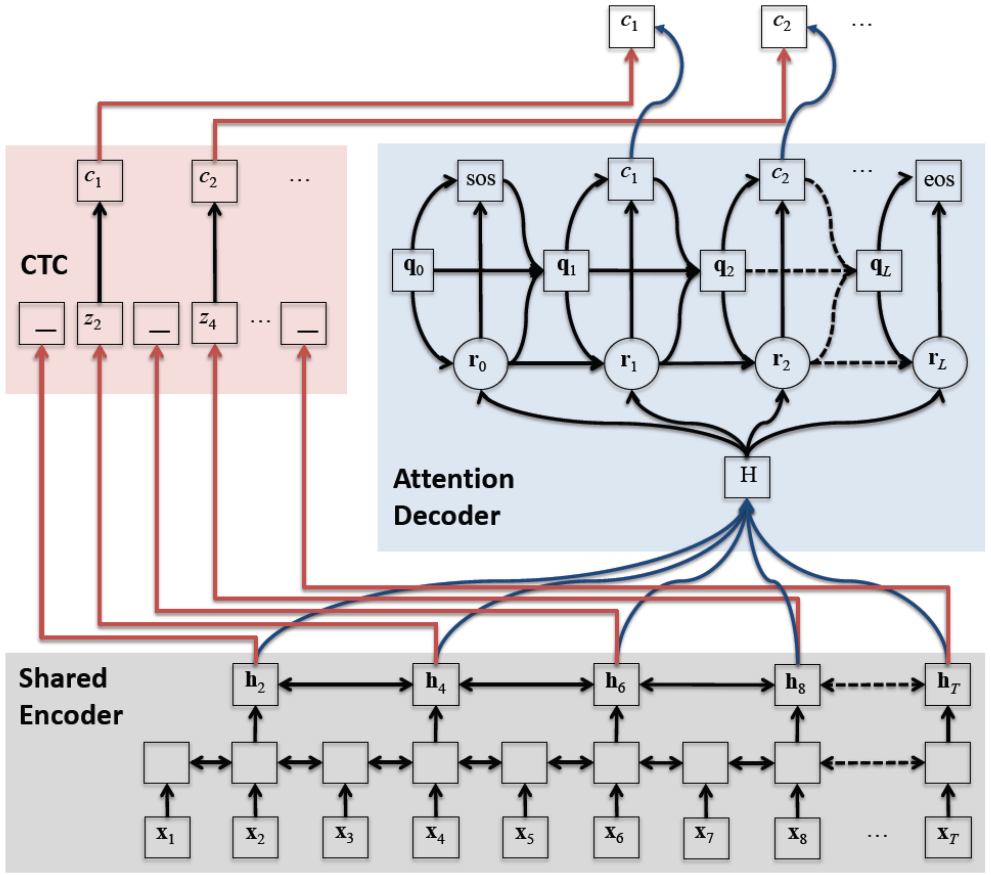
(c.) Attention-based Model



(d.) RNN-Transducer with Attention

Hybrid CTC/Seq2seq

1. S. Watanabe, T. Hori, S. Kim, J. Hershey, T. Hayashi, Hybrid CTC/Attention Architecture for End-to-End Speech Recognition. IEEE Journal of Selected Topics in Signal Processing, Volume:11 , Issue: 8, pp 1240-1253

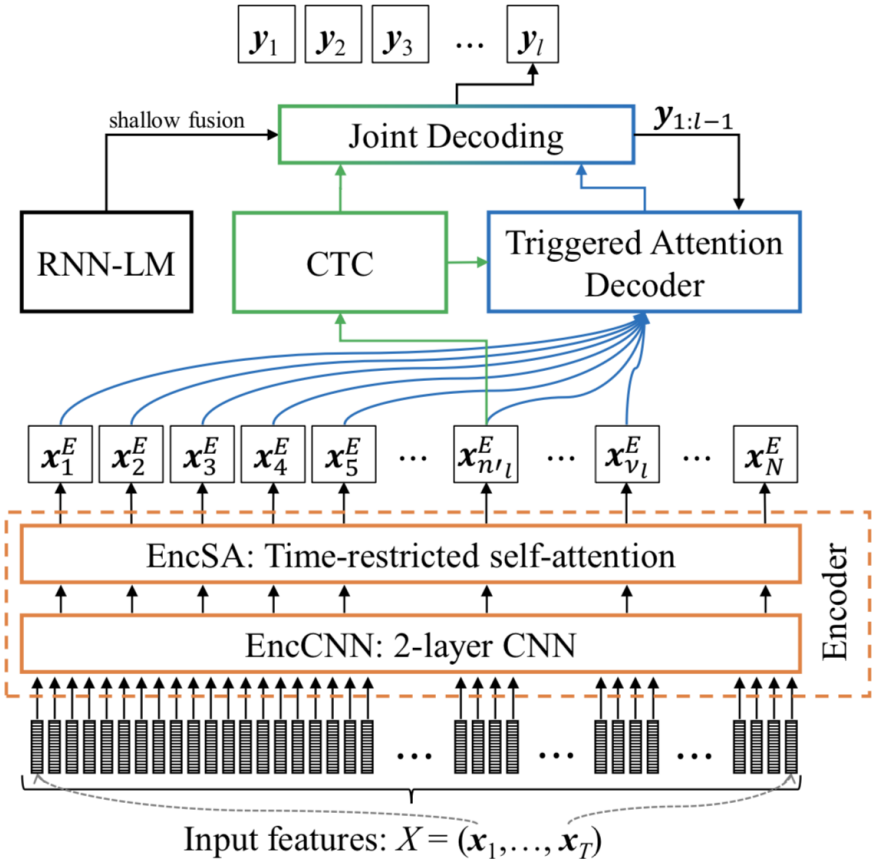


Algorithm 1 Joint CTC/attention one-pass decoding

procedure ONEPASSBEAMSEARCH(X, $L_{\max}$)
  $\Omega_0 \leftarrow \{\texttt{<sos>}\}$
  $\hat{\Omega} \leftarrow \emptyset$
  for $l = 1 \ldots L_{\max}$ **do**
    $\Omega_l \leftarrow \emptyset$
    while $\Omega_{l-1} \neq \emptyset$ **do**
      $g \leftarrow \text{HEAD}(\Omega_{l-1})$
      DEQUEUE(Ω_{l-1})
      for each $c \in \mathcal{U} \cup \{\texttt{<eos>}\}$ **do**
        $h \leftarrow g \cdot c$
        $\alpha(h) \leftarrow \lambda\alpha_{\text{ctc}}(h, X) + (1-\lambda)\alpha_{\text{att}}(h, X)$
        if $c = \texttt{<eos>}$ **then**
          ENQUEUE($\hat{\Omega}, h$)
        else
          ENQUEUE(Ω_l, h)
          if $|\Omega_l| > beamWidth$ **then**
            REMOVEWORST(Ω_l)
          end if
        end if
      end for
    end while
    if ENDDETECT($\hat{\Omega}, l$) = `true` **then**
      break ▷ exit for loop
    end if
  end for
  return $\arg\max_{C \in \hat{\Omega}} \alpha(C)$
end procedure

Trigged attention

- N. Moritz, T. Hori, J. Roux. Streaming Automatic Speech Recognition with the Transformer Model. arXiv:2001,02674. 2020



Algorithm 1 Joint CTC-triggered attention decoding

procedure DECODE(X_E, p_{ctc}, λ, α_0, α, β, K, P, θ_1, θ_2)
  $\ell \leftarrow (\langle \text{sos} \rangle,)$
  $\Omega \leftarrow \{\ell\}$, $\Omega_{\text{ta}} \leftarrow \{\ell\}$
  $p_{\text{nb}}(\ell) \leftarrow 0$, $p_{\text{b}}(\ell) \leftarrow 1$
  $p_{\text{ta}}(\ell) \leftarrow 1$
  for $n = 1, \ldots, N$ **do**
    $\Omega_{\text{ctc}}, p_{\text{nb}}, p_{\text{b}} \leftarrow$ CTCPREFIX($p_{\text{ctc}}(n), \Omega, p_{\text{nb}}, p_{\text{b}}$)
    for ℓ **in** Ω_{ctc} **do** ▷ Compute CTC prefix scores
      $p_{\text{prfx}}(\ell) \leftarrow p_{\text{nb}}(\ell) + p_{\text{b}}(\ell)$
      $\widehat{p}_{\text{prfx}}(\ell) \leftarrow \log p_{\text{prfx}}(\ell) + \alpha_0 \log p_{\text{LM}}(\ell) + \beta|\ell|$
    $\widehat{\Omega} \leftarrow$ PRUNE($\Omega_{\text{ctc}}, \widehat{p}_{\text{prfx}}, K, \theta_1$)
    for ℓ **in** $\widehat{\Omega}$ **do** ▷ Delete old prefixes in Ω_{ta}
      if ℓ **in** Ω_{ta} **and** DCOND($\ell, \widehat{\Omega}, p_{\text{ctc}}$) **then**
        delete ℓ in Ω_{ta}
    for ℓ **in** $\widehat{\Omega}$ **do** ▷ Compute transformer scores
      if ℓ **not in** Ω_{ta} **and** ACOND($\ell, \widehat{\Omega}, p_{\text{ctc}}$) **then**
        $p_{\text{ta}}(\ell) \leftarrow$ DECTA($x^E_{1:n+\varepsilon^{\text{dec}}}, \ell$)
        add ℓ to Ω_{ta}
    for ℓ **in** $\widehat{\Omega}$ **do** ▷ Compute joint scores
      $\widehat{\ell} \leftarrow \ell$ **if** ℓ **in** Ω_{ta} **else** $\ell_{:-1}$
      $p \leftarrow \lambda \log p_{\text{prfx}}(\ell) + (1 - \lambda) \log p_{\text{ta}}(\widehat{\ell})$
      $p_{\text{joint}}(\ell) \leftarrow p + \alpha \log p_{\text{LM}}(\ell) + \beta|\ell|$
    $\Omega \leftarrow$ MAX($\widehat{\Omega}, p_{\text{joint}}, P$)
    $\widehat{\Omega} \leftarrow$ PRUNE($\widehat{\Omega}, \widehat{p}_{\text{prfx}}, P, \theta_2$)
    $\Omega \leftarrow \Omega + \widehat{\Omega}$
    remove from Ω_{ta} prefixes rejected due to pruning
  return MAX($\widehat{\Omega}, p_{\text{joint}}, 1$)

小结

深蓝学院
shenlanxueyuan.com

2 语音识别系统与解码器

- 2.1 动态解码器
- 2.2 WFST解码器
- 2.3 Seq2seq解码过程
- 2.4 几种端到端语音识别系统对比

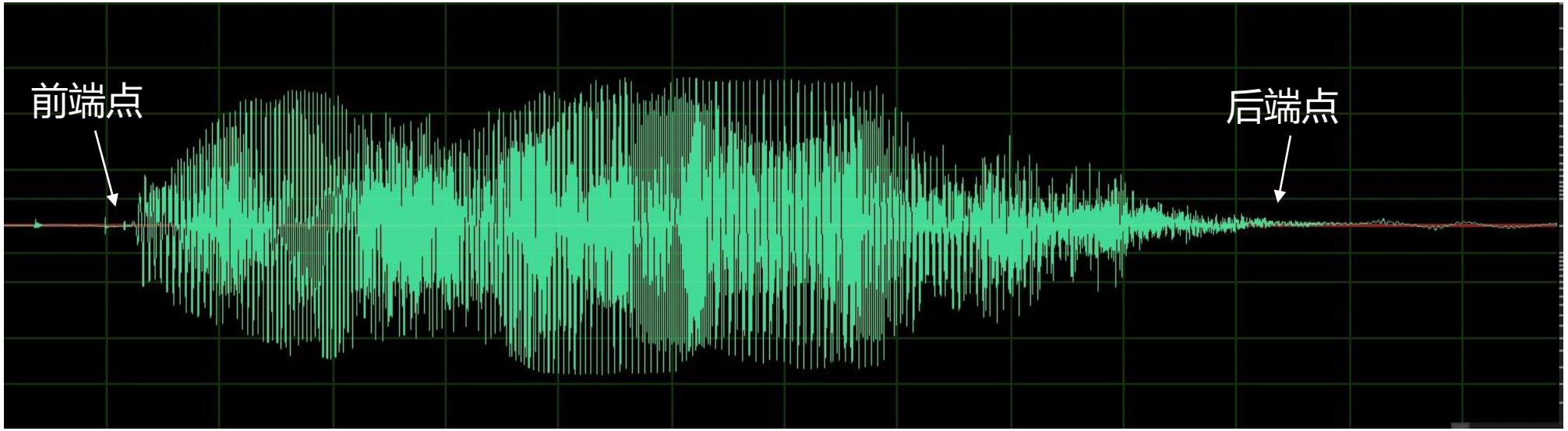
搭建语音识别系统

深蓝学院 shenlanxueyuan.com

3 Voice Activity Detector (VAD)

深蓝学院 shenlanxueyuan.com

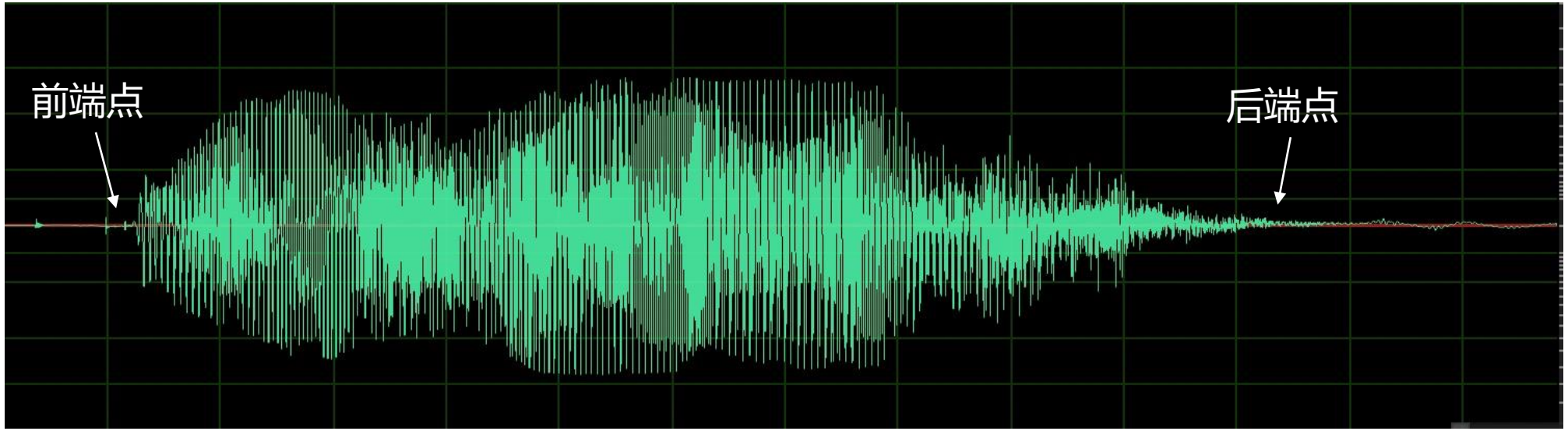
- VAD (端点检测)
  - 从语音信号中将语音（Speech）和非语音（Nonspeech）区分开，确定语音信号的端点，包括前端点和后端点



3 Voice Activity Detector (VAD)

深蓝学院 shenlanxueyuan.com

- 为什么VAD很重要
  - 太灵敏 or 太迟钝



3.1 能量VAD

深蓝学院 shenlanxueyuan.com

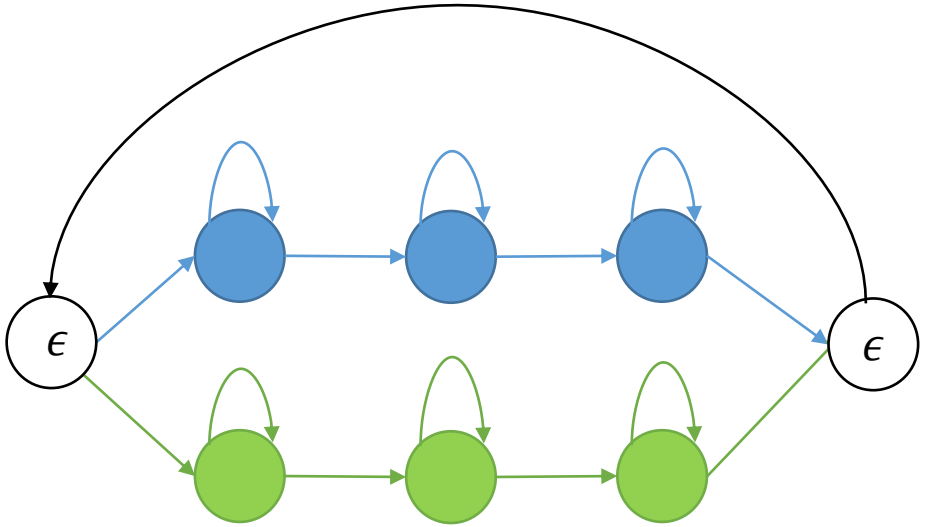
- 基于特征的方法
  - 采用能对语音和非语音（噪声）具有区分度的特征判断
  - 常用特征：能量，过零率，基频等

```
namespace kaldi {

void ComputeVadEnergy(const VadEnergyOptions &opts,
                      const MatrixBase<BaseFloat> &feats,
                      Vector<BaseFloat> *output_voiced) {
  int32 T = feats.NumRows();
  output_voiced->Resize(T);
  if (T == 0) {
    KALDI_WARN << "Empty features";
    return;
  }
  Vector<BaseFloat> log_energy(T);
  log_energy.CopyColFromMat(feats, 0); // column zero is log-energy.

  BaseFloat energy_threshold = opts.vad_energy_threshold;
  if (opts.vad_energy_mean_scale != 0.0) {
    KALDI_ASSERT(opts.vad_energy_mean_scale > 0.0);
    energy_threshold += opts.vad_energy_mean_scale * log_energy.Sum() / T;
  }
  KALDI_ASSERT(opts.vad_frames_context >= 0);
  KALDI_ASSERT(opts.vad_proportion_threshold > 0.0 &&
               opts.vad_proportion_threshold < 1.0);
  for (int32 t = 0; t < T; t++) {
    const BaseFloat *log_energy_data = log_energy.Data();
    int32 num_count = 0, den_count = 0, context = opts.vad_frames_context;
    for (int32 t2 = t - context; t2 <= t + context; t2++) {
      if (t2 >= 0 && t2 < T) {
        den_count++;
        if (log_energy_data[t2] > energy_threshold)
          num_count++;
      }
    }
    if (num_count >= den_count * opts.vad_proportion_threshold)
      (*output_voiced)(t) = 1.0;
    else
      (*output_voiced)(t) = 0.0;
  }
}
```

3.2 基于HMM的VAD

深蓝学院 shenlanxueyuan.com

- 将VAD看做是一个特殊的语音识别任务
  - 其发音词典（或者声学模型）只有Silence和Speech
- 训练方法
  - 1）基于语音识别的Alignment得到每一帧特征对应的声学单元
  - 2）将非silence部分统一设置为speech
  - 3）基于EM算法训练GMM 或者DNN
- 一些小优化
  - 1）将Speech部分采用更多声学单元建模
- 基于DNN的VAD的两种框架
  - HMM框架、得分加窗平滑的方案



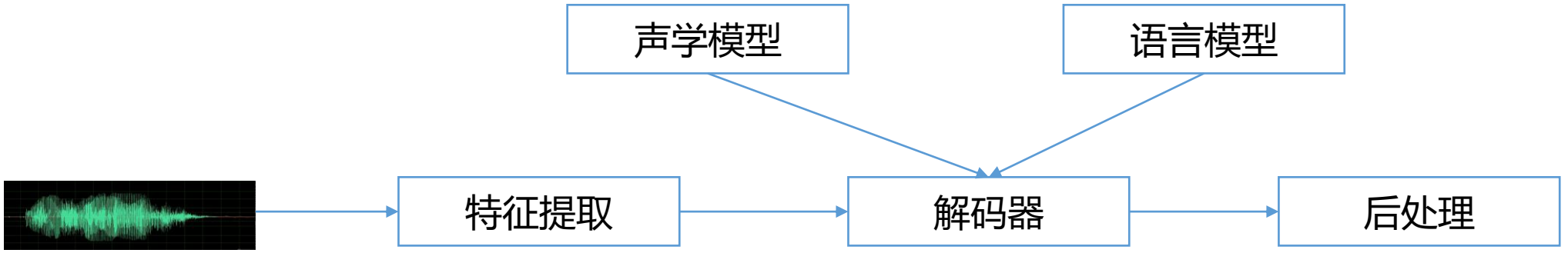
VAD的一些改进

- VAD与语音识别过程结合
  - 在语音识别声学建模过程中，将后端点（endpoint）的检测和声学模型一起联合建模
  - S. Chang, R. Prabhavalkar, Y. He, T. Sainath. Joint Endpointing and Decoding with End-to-End Models. ICASSP 2019
- VAD与语义理解结合
  - 基于文字内容判断一段语音识别说完整
  - 结合语音识别结果以及声学信号，训练分类模型
- 更小的模型
  - Binary Neural Network ...

搭建语音识别系统

深蓝学院 shenlanxueyuan.com

4 讨论 - 搭建语音识别系统



- Streaming语音识别：VAD
- 语音识别解码器的评估指标之一：RTF（Real Time Factor）

云端语音识别 vs 嵌入式语音识别

深蓝学院 shenlanxueyuan.com

- 嵌入式语音识别
  - 核心目标：用相对较少的计算资源实现相对较好的识别性能
  - 基于发音词典构建解码网络
    - 发音词典的大小（声学和语言建模单元）-> 解码网络的大小
  - 通过声学模型实现声学到发音单元的识别
    - 声学模型通常为神经网络 -> 识别系统的计算量
  - 通过语言模型消歧同音序列
    - 语言模型通常采用Ngram -> 识别系统的内存占用

云端语音识别 vs 嵌入式语音识别

深蓝学院 shenlanxueyuan.com

- 云端语音识别
  - Streaming 语音识别中，哪些模块需要放在设备端上？
    - VAD：一般情况下，在设备端做一级VAD判断，在云端做二级判断
    - 特征提取：一般情况下，在云端做特征提取，在设备端压缩语音上传
  - 并发与实时
    - Streaming 语音识别要求当录音上传到云端时候，一定有计算线程空闲支持识别
    - Offline 语音识别可以整体调度

场景自适应

深蓝学院 shenlanxueyuan.com

- “离开场景谈语音识别性能就是耍流氓”
- 常见场景自适应的方式
  - 定位场景差异：声学 or 语言
  - 语言模型 -> 声学模型
- 语言模型自适应

  1）基于提供的文本自动训练Ngram，得到领域语言模型

  2）将领域语言模型与通用语言模型插值

  3）快速发布上线

后处理

深蓝学院 shenlanxueyuan.com

- 后处理通常做什么
  - 关键词修正
  - 结合场景或者语义信息实现语音识别结果纠错
  - 加标点符号，提升可读性
- 识别结果纠错方法举例
  - 基于Seq2seq模型，训练纠错模型
    - 输入为识别结果，输出为标注正确答案
- 标点预测方案举例
  - 基于Ngram，采用WFST实现标点预测
  - 常见序列标注算法：如最大熵，CRF，...

搭建语音识别系统

深蓝学院 shenlanxueyuan.com

5 课后作业

深蓝学院 shenlanxueyuan.com

- 基本要求：查询文献，论述以下问题，形成文档
  - 问题1：基于WFST实现语音文字的强制对齐（Force Alignment）
    - 如何基于一句话构建其WFST $G_{sentence}$
    - 标准的CTC模型为什么不适合做对齐，那么如何修改CTC配置实现对齐?
  - 问题2：基于WFST实现基于N-Gram语言模型的序列标注/转换任务
    - 拼音输入法
    - 分词、注音、加标点

5 课后作业

深蓝学院 shenlanxueyuan.com

- 进阶要求：基于Openfst工具实现其中一个任务，自选训练数据基于KenLM训练Ngram模型，并形成文档描述实现方式即可（不用关注性能）
  - 拼音输入法
  - 注音
  - 加标点
- Note:
  - 安装并使用Kenlm：https://github.com/kpu/kenlm
    - 参考kaldi里面的工具：kaldi/lmbin/arpa2fst 将ngram转为fst
  - 安装并使用openfst：http://www.openfst.org/
    - conda install –c conda-forge openfst
    - 常用工具：fstcompose, fstshortestpath, …

深蓝学院
shenlanxueyuan.com

谢谢！