语言模型

洪青阳

纲要

8.1 n-gram模型

8.2 评价指标—Perplexity

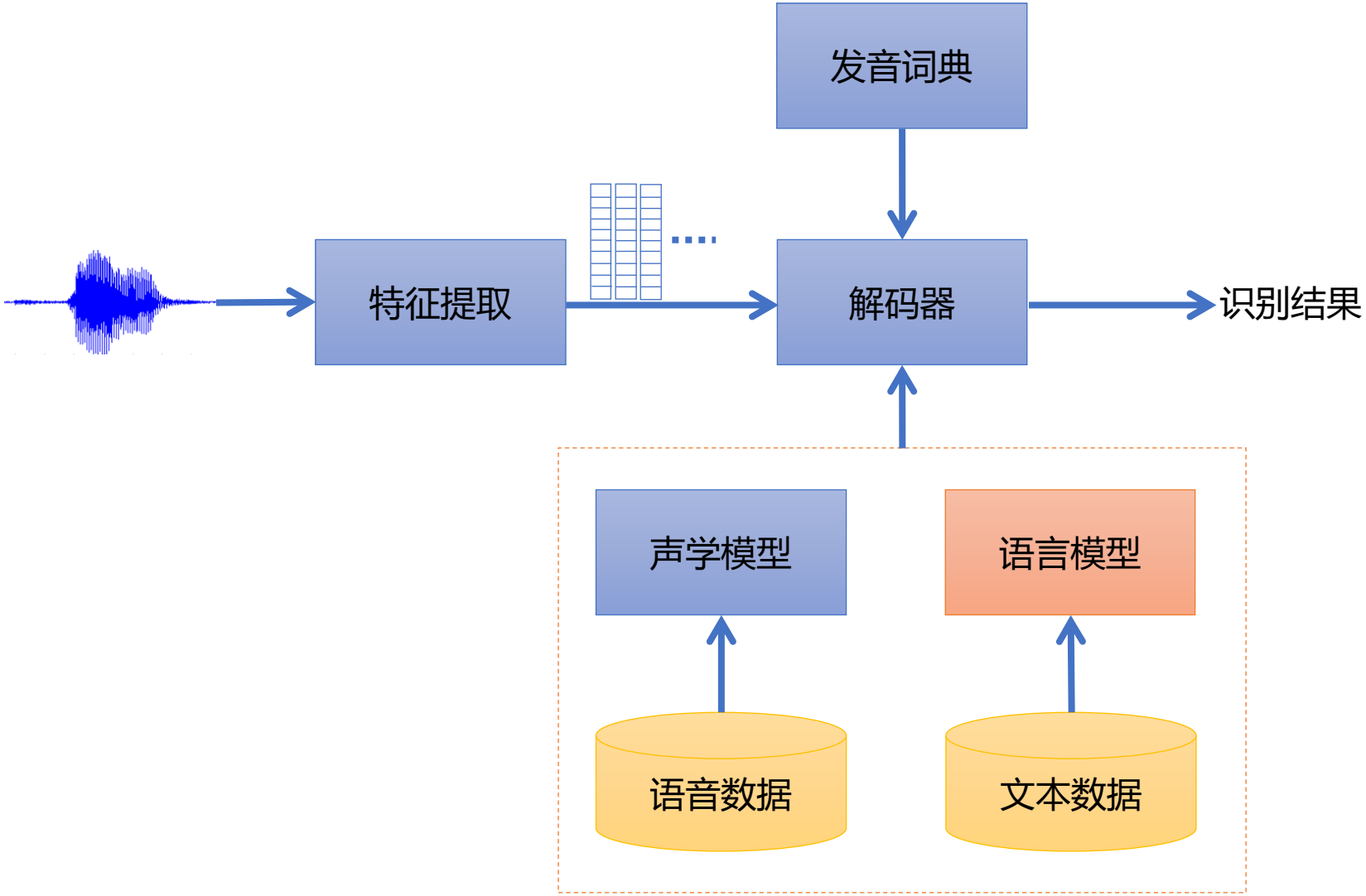
8.3 平滑技术

8.4 语言模型的训练

8.5 神经网络语言模型（NNLM）

8.6 本章小结

8.1 n-gram模型



8.1 n-gram模型

- 语音识别的任务为，找到对应观察序列O的最可能的词序列$\widehat{W}$。按贝叶斯准则：

$$\widehat{W} = arg\max_{W} P(W|O) = arg\max \frac{P(W)P(O|W)}{P(O)}$$

$$\widehat{W} = arg\max_{W} P(W)P(O|W)$$

- 要找到最可能的词序列，必须使上式右侧两项的乘积最大。其中$P(O|W)$由声学模型决定，$P(W)$由语言模型决定。

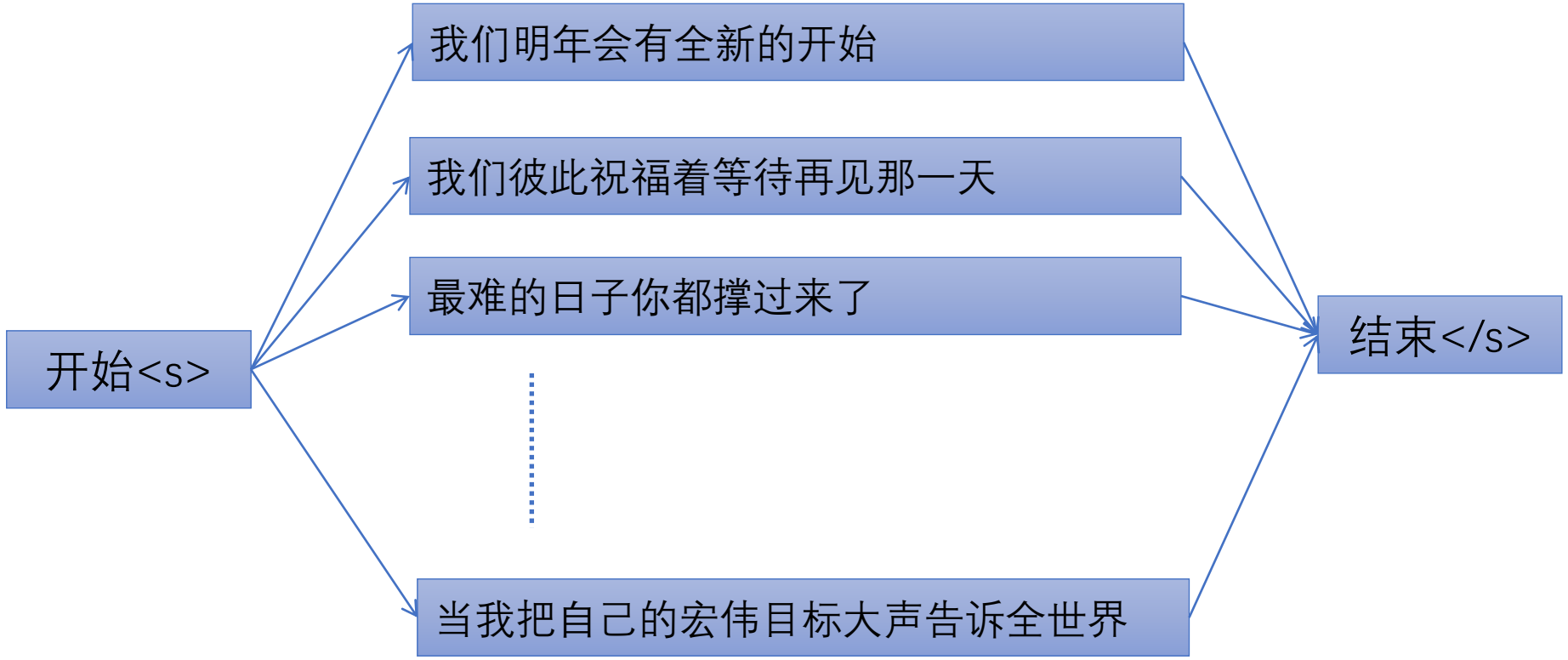
- 语言模型用来表示词语序列出现的可能性，可以基于语法规则，也可以基于统计方法。

8.1 n-gram模型

- 基于规则的语言模型来源于语言学家掌握的语言学知识和领域知识，或者根据特定应用设定语法规则，一般仅能约束受限领域内的句子。

- 统计语言模型，通过对大量文本语料进行处理，获取给定词序列的概率分布，从而能够客观描述隐含的规律，适合于处理大规模真实文本。统计语言模型已被广泛应用于语音识别、机器翻译、文本校对等多个领域。

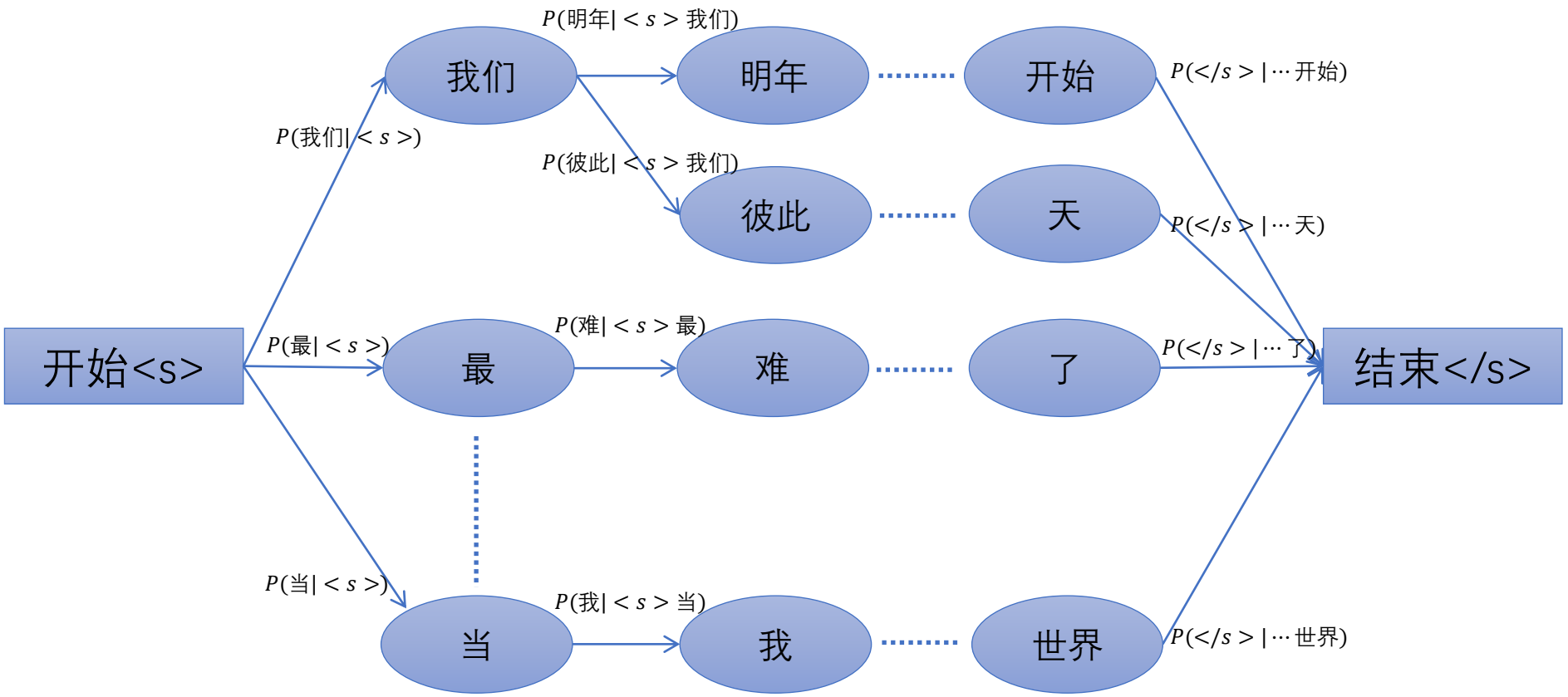
8.1 n-gram模型



不同句子：各种词序列

8.1 n-gram模型

$$P(w_1 \cdots w_{t-1} w_t)$$
$$= P(w_1)P(w_2|w_1)P(w_3|w_1 w_2) \cdots P(w_t|w_{t-n+1} w_{t-n+2} \cdots w_{t-1})$$



词与词之间的组合概率关系

8.1 n-gram模型

统计语言模型的分类:

- n-gram
  - 预测的词概率值依赖于前n-1个词，更长距离上下文依赖被忽略。
  - 效率高，目前仍是主流!

- 神经网络（RNN/LSTM/Transformer）
  - RNN/LSTM将每个词映射到一个紧凑的连续向量空间，并使用循环连接来建模长距离上下文依赖。
  - 输出层的节点数目就是词典的大小，处理大量数据时缓慢的训练速度限制了RNN/LSTM的使用。
  - 也可采用Transformer建模。

8.1 $\boldsymbol{n}$-gram模型

- 1980年由Fred Jelinek和他的同事提出n-gram统计语言模型。每个预测变量w_t只与长度为$n-1$的上下文有关：

$$P(w_t|w_1 \cdots w_{t-1}) = P(w_t|w_{t-n+1}w_{t-n+2} \cdots w_{t-1})$$

- 实际应用中一般取$1 \leq n \leq 5$。当$n = 1,2,3$时，相应的模型分别称为Unigram, Bigram, Trigram语言模型。

8.1 n-gram模型

- n-gram 概率从文本语料估计得到，可简单地计算词的出现个数。

- 文本语料:

  “我们明年会有全新的开始”
  “我们彼此祝福着等待再见那一天”
  “最难的日子你都撑过来了”
  ……
  “当我把自己的宏伟目标大声告诉全世界”

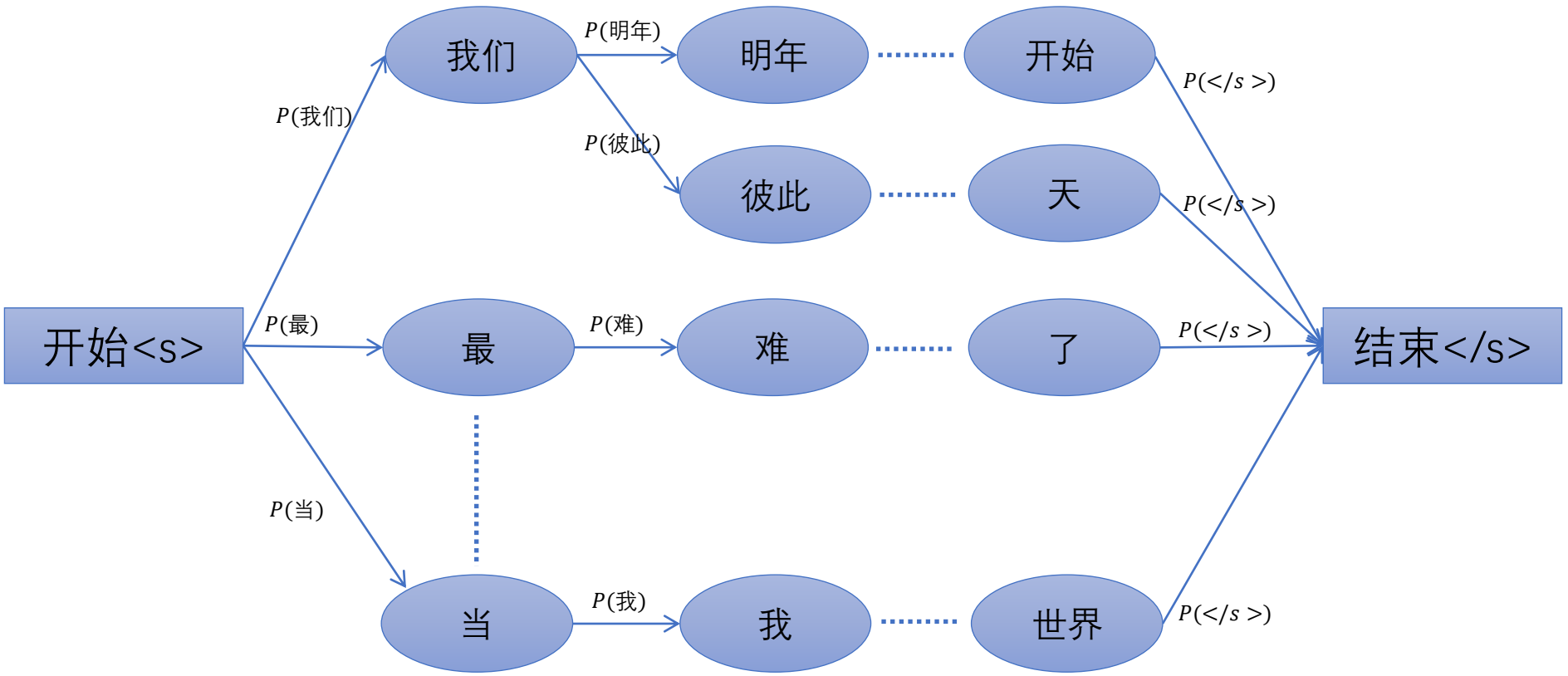
1-gram的训练：Unigram

- 假设有1000句语料，总共有20000个词
  - “我们”出现100次，“明年”出现30次，“日子”出现10次，……
  - 总共有21000个词标签，其中包括1000句的结束符</s>。

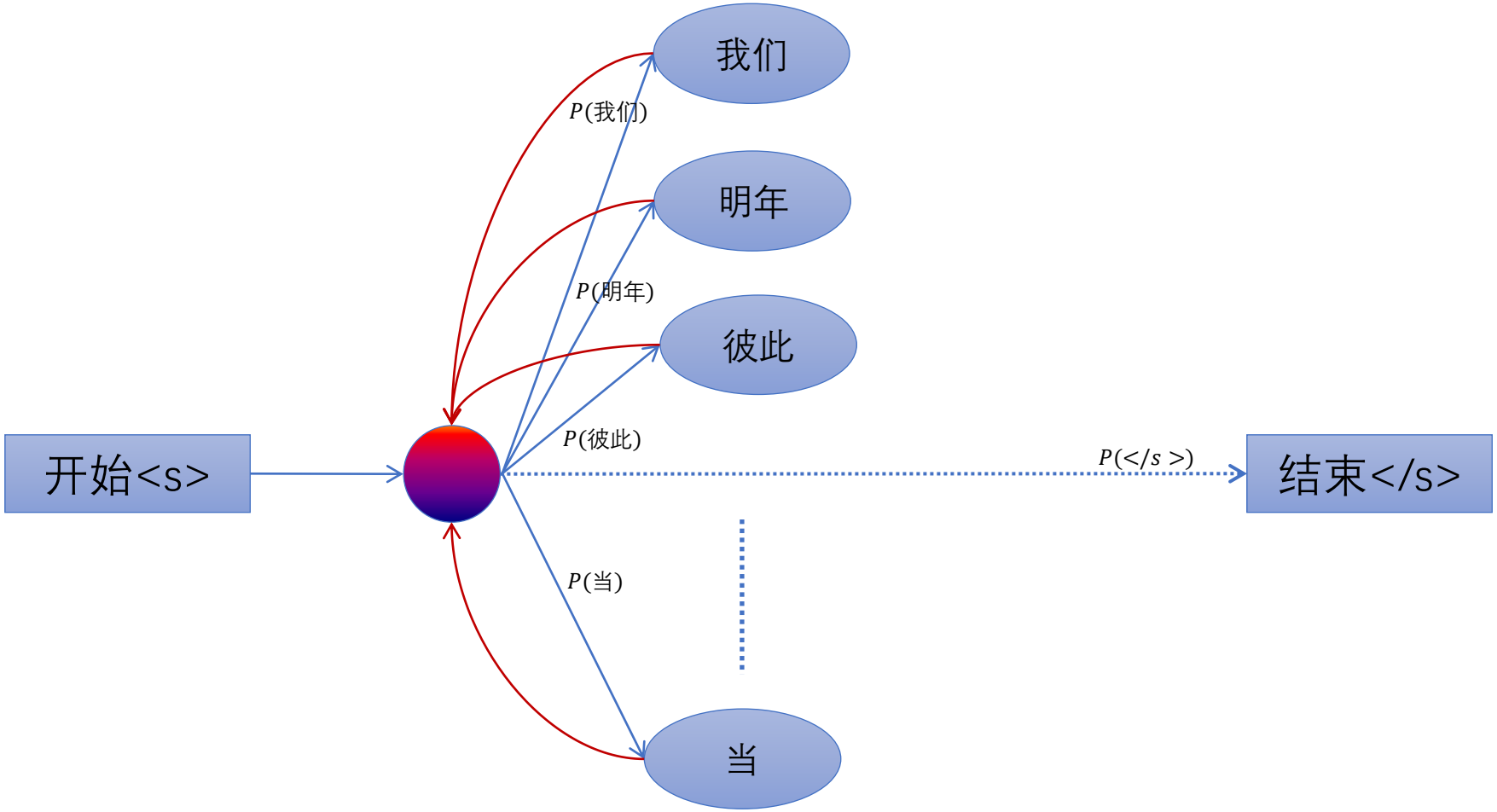
- Unigram计算：
  - P(“我们”) = 100/21000
  - P(“明年”) = 30/21000
  - P(“日子”) = 10/21000
  - P(</s>) = 1000/21000

我们	明年	日子	会	……	世界
100	30	10	8	……	3

1-gram语言模型：Unigram



1-gram语言模型：Unigram

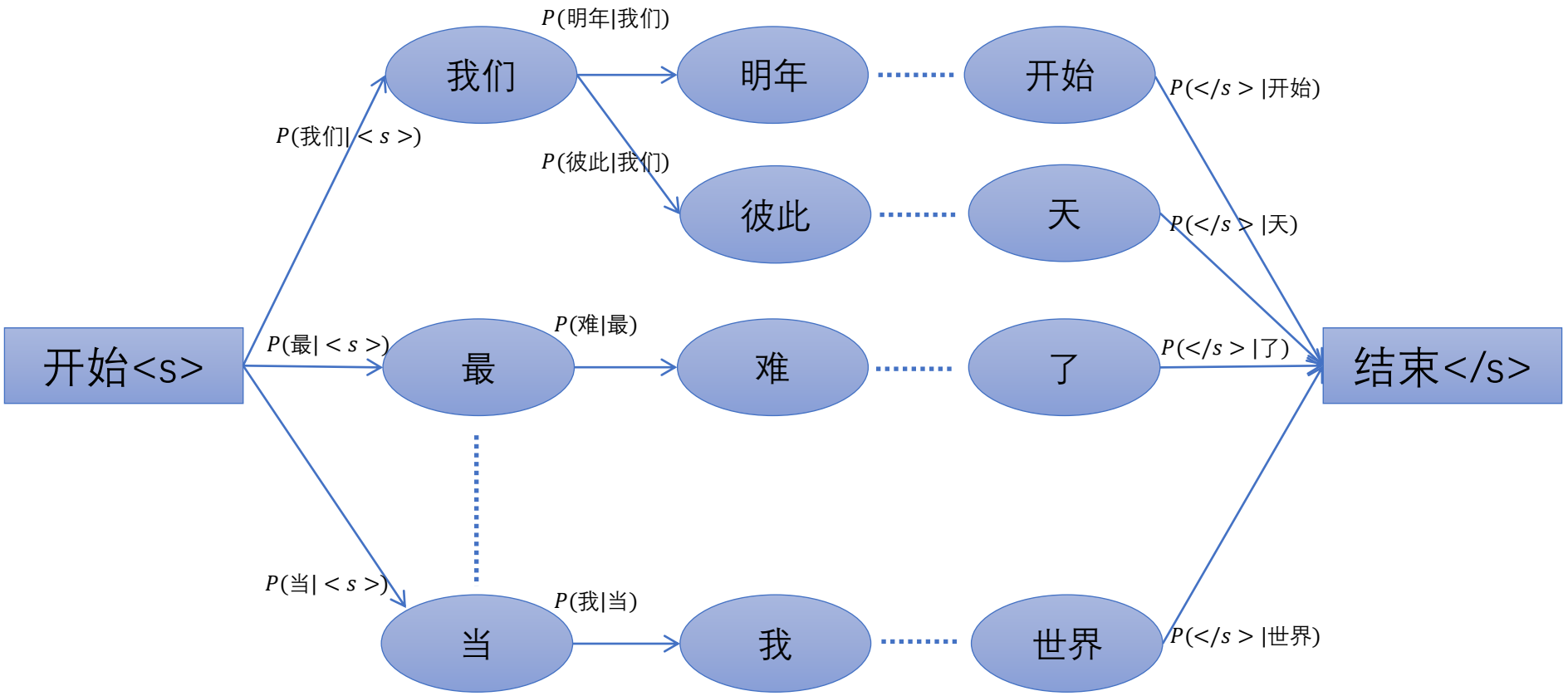


2-gram的训练：Bigram

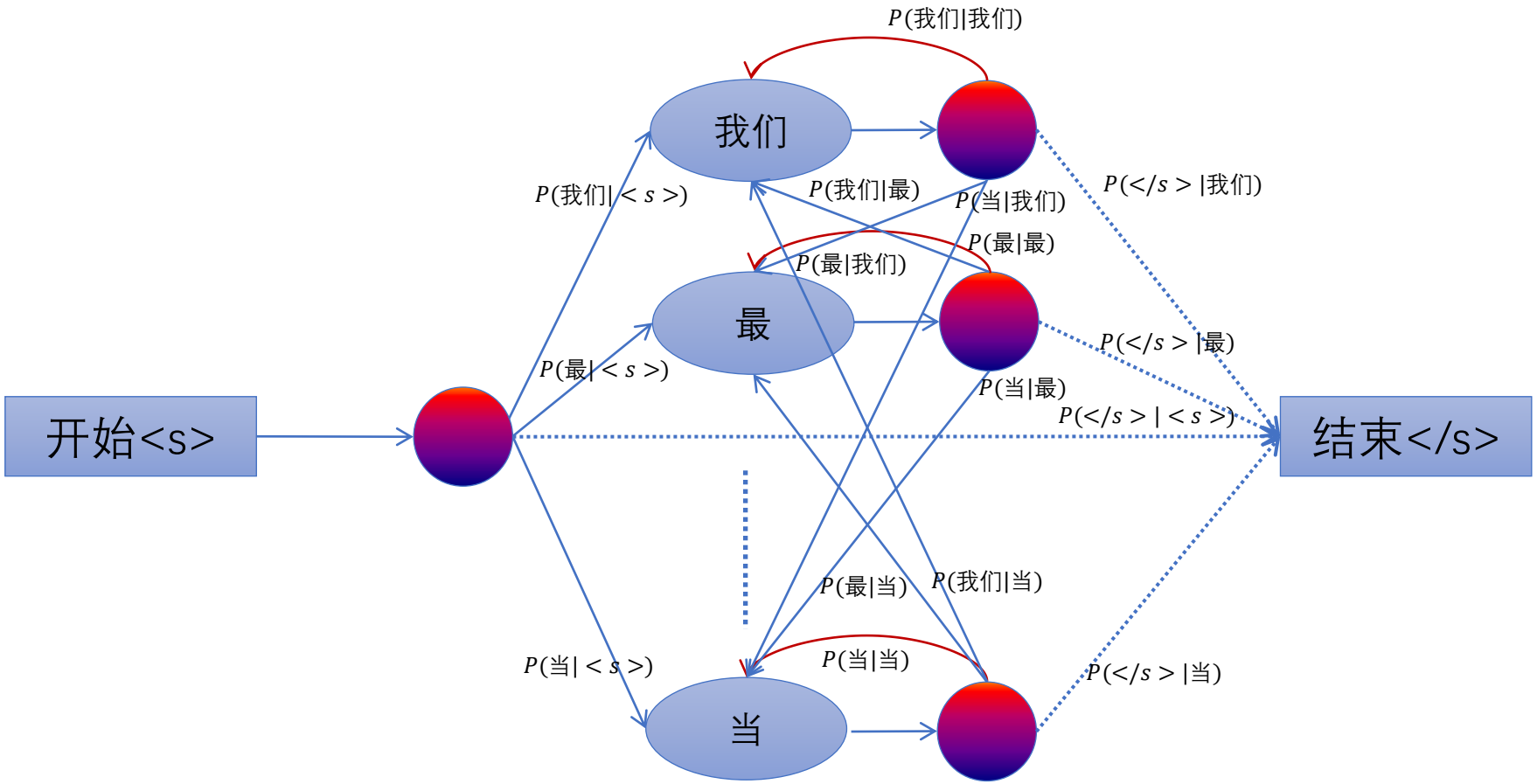
- 假设1000句语料中出现：
  - 10句以“我们”开头，5句以“明天”开头，……
  - 2句以“日子”结尾，……
  - 1次出现“我们明年”，3次出现“我们彼此”，……
- 则Bigram计算如下：
  - P(“我们”|<s>) = 10/1000
  - P(“明天”|<s>) = 5/1000
  - P(</s>| “日子”) = 2/10 ---“日子”出现10次
  - P(“明年”|“我们”) = 1/100 ---“我们”出现100次
  - P(“彼此”|“我们”) = 3/100

	我们	明天	日子	明年	彼此
<s>	0.01	0.005	0	0	0
我们	0	0	0	0.01	0.03
明天	0	0	0	0	0

2-gram语言模型： Bigram



2-gram语言模型： Bigram



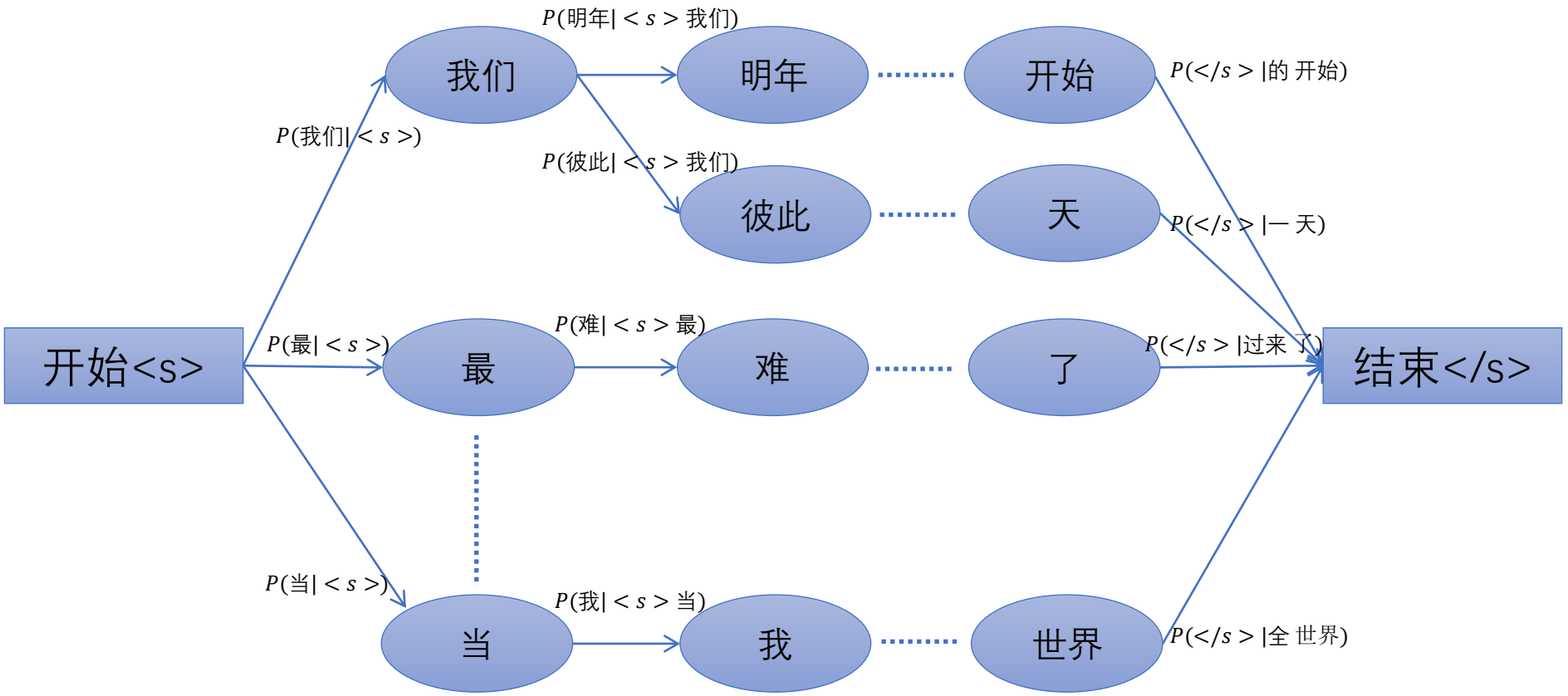
3-gram的训练：Trigram

- 概率计算

$$P(w_3|w_1w_2) = count(w_1w_2w_3)/count(w_1w_2)$$

- 假设“我们 明天”出现2次，“我们 明天 开始”出现1次，则 P(“开始”|“我们 明天”) = 1/2

3-gram语言模型：Trigram



8.2 评价指标—Perplexity

- 给定句子S，包含词序列$w_1 \cdots w_T$，T是句子长度，则Perplexity表示为：

$$\mathrm{PPL}(W) = P(w_1 w_2 \ldots w_T)^{-\frac{1}{T}} = \sqrt[T]{\frac{1}{P(w_1 w_2 \ldots w_T)}}$$

- Perplexity又称**困惑度(PPL)**，PPL越小，$P(w_i)$则越大，句子S出现的概率就越高。

- 理论上，困惑度越小，语言模型越好，预测能力越强。

n-gram	Unigram	Bigram	Trigram
Perplexity	9568	2374	1686

8.3 平滑技术

- 统计语料有限，存在数据稀疏，导致零概率问题。

- 平滑方法
  - - **折扣法(Discounting)**：从已有观察概率调配一点给未观察概率，例如Good-Turing折扣法、Witten-Bell折扣法。
  - - **回退法(Back-off)**：基于低阶模型估计未观察到的高阶模型，例如Katz回退法。
  - - **插值法(Interpolation)**：将高阶模型和低阶模型做线性组合，如Jelinek-Mercer插值法，也可非线性组合，如Kneser-Ney插值法。

Good-Turing折扣法

- 算法：
  - 设总词数为N，平滑前出现1次的词数为N_1，出现c次的词数为N_c。
  - 平滑后，概率P^*(出现0次的词) $= \frac{N_1}{N}$，出现次数$c^* = \frac{(c+1)N_{c+1}}{N_c}$。
- 例子：
  - 分词后句子语料（假设只有2句）：
    - “我们 明年 会 有 全新 的 开始”
    - “我们 彼此 祝福 着 等待 再见 那 一 天”
  - 词频数：“我们”2次，“明年”1次，……，“天”1次
  - 折扣前： $N = 16, N_1 = 14$, $N_2 = 1$
  - 折扣后： $c_1^* = \frac{(1+1)N_2}{N_1} = \frac{2}{14}$。为标识清楚，这里用$c_1^*$表示出现1次的折扣后次数。因为$N_3 = 0$，根据折扣公式，则$c_2^*$也变为0。
  - 折扣后概率：$P_0^* = \frac{N_1}{N} = \frac{14}{16}$，$P_1^* = \frac{c_1^*}{N} = \frac{2}{14*16}$

Good-Turing折扣法（2-gram）

在实际应用中，出现次数太少的词组合并不可靠，假定这种组合最小次数为$gtmin$，最大次数为$gtmax$，一种改进办法是把$gtmin < C < gtmax$的n-gram削减掉，并采用校正因子，分配给未出现的n-gram。

1）Good-Turing新计数

$$C^*(w_{t-1}w_t) = (C(w_{t-1}w_t) + 1)\frac{N_{(C(w_{t-1}w_t)+1)}}{N_{C(w_{t-1}w_t)}}$$

2）校正因子A

$$A = (gtmax + 1)\frac{N_{(gtmax+1)}}{N_1}$$

3）折扣率d

$$d = \frac{\frac{C^*(w_{t-1}w_t)}{C(w_{t-1}w_t)} - A}{1 - A}$$

4）最大似然估计

$$P(w_t|w_{t-1}) = \begin{cases} \frac{C(w_{t-1}w_t)}{C(w_{t-1})}, & C(w_{t-1}w_t) > gtmax \\ d\frac{C(w_{t-1}w_t)}{C(w_{t-1})}, & gtmin \le C(w_{t-1}w_t) \le gtmax \text{且} d > 0 \\ \frac{C(w_{t-1}w_t)}{C(w_{t-1}) + 1}, & d < 0 \end{cases}$$

利用上述四个公式来计算 2-gram。当$d < 0$时，采用简化计算公式，即分母加1。

Witten-Bell折扣法（针对语料太少）

- 2-gram

$$P(w_t|w_{t-1}) = \frac{C(w_{t-1}w_t)}{C(w_{t-1}_)+n[w_{t-1}_*]},$$ 其中$n[w_{t-1}_*]$代表w_{t-1}后接不同词的类数。

例如：

- “我们 明年 会 有 全新 的 开始”
- “我们 彼此 祝福 着 等待 再见 那 一 天”

包含“我们 明年”和“我们 彼此”各一次，即C(“我们 明年”) $= 1$，C(“我们 彼此”) $= 1$，则C[“我们_”] $= 2$，n[“我们_ * ”] $= 2$

假如“我们 明年”多出现一次，则：C[“我们_”] $= 3$，n[“我们_ * ”] $= 2$。

- 更通用写法

$$P(\mathrm{a_}z) = \frac{C(a_z)}{C(a_)+n[a_*]},$$ 其中$n[a_*]$代表a后接不同词的类数。

Katz回退法

- 采用Katz平滑技术的概率估计公式如下：

$$P(w_t|w_{t-2}w_{t-1}) = \begin{cases} \dfrac{C(w_{t-2}w_{t-1}w_t)}{C(w_{t-2}w_{t-1})}, \text{当} C > C' \\ d\dfrac{C(w_{t-2}w_{t-1}w_t)}{C(w_{t-2}w_{t-1})}, \text{当} 0 < C < C' \\ backoff(w_{t-2}w_{t-1})P(w_t|w_{t-1}) \end{cases}$$

其中C是$C(w_{t-2}w_{t-1}w_t)$的简写，表示三个词同时出现的次数，C'是一个计数阈值，当$C > C'$时，直接采用最大似然方法估计概率；当$0 < C < C'$时，则采用Good-Turing折扣法，d是折扣系数。

$backoff(w_{t-2}w_{t-1})$是回退权重，回退权重的计算如下：

$$backoff(w_{t-2}w_{t-1}) = \frac{1 - \sum_{w_t} P(w_t|w_{t-2}w_{t-1})}{1 - \sum_{w_t} P(w_t|w_{t-1})}$$

其中w_t是在训练语料中$w_{t-2}w_{t-1}$之后出现的词。

Bigram计算过程(Good-Turing+Katz回退)

1）Good-Turing新计数

$$C^*(w_{t-1}w_t) = (C(w_{t-1}w_t)+1)\frac{N_{(C(w_{t-1}w_t)+1)}}{N_{C(w_{t-1}w_t)}}$$

2）校正因子A

$$A = (gtmax+1)\frac{N_{(gtmax+1)}}{N_1}$$

3）折扣率d

$$d = \frac{\frac{C^*(w_{t-1}w_t)}{C(w_{t-1}w_t)} - A}{1-A}$$

4）Katz回退权重

$$backoff(w_{t-1}) = \frac{1-\sum_{w_t} P(w_t|w_{t-1})}{1-\sum_{w_t} P(w_t)}$$

5）最大似然估计

$$P(w_t|w_{t-1}) = \begin{cases} \frac{C(w_{t-1}w_t)}{C(w_{t-1})}, & C(w_{t-1}w_t) > gtmax \\ d\frac{C(w_{t-1}w_t)}{C(w_{t-1})}, & gtmin \le C(w_{t-1}w_t) \le gtmax \text{且} d > 0 \\ \frac{C(w_{t-1}w_t)}{C(w_{t-1})+1}, & gtmin \le C(w_{t-1}w_t) \le gtmax \text{且} d < 0 \\ backoff(w_{t-1})P(w_t) & \end{cases}$$

当分子为0时，回退权重就为0，即回退概率为0，意味着我们看到了包含这个前缀的所有的组合，然而训练的文本不可能包含所有的组合，所以我们需要给回退概率分配一定的值，这时就需要对之前计算的概率进行分母修正，使得概率小一点，这样概率之和小于1，回退权重就会大于0，回退概率也会大于0。

注意：在计算$backoff$的时候，如果分子为0，对相关的概率值计算使分母+1，并修改对应的n-gram的似然估计值（分母+1）。

针对一元到五元模型，$gtmin$默认值可设置为1、1、2、2、2，$gtmax$则为1、7、7、7、7。当$N_{(gtmax+1)}$为0时，可再减小$gtmax$，直至$N_{(gtmax+1)} > 0$。

英文计算例子

dogs chase cats

dogs bark

cats meow

dogs chase birds

cats chase birds

dogs chase the cats

the birds chirp

$gtmax$默认为7，$N_{(gtmax+1)}$为0

减小$gtmax$到3，$N_{(gtmax+1)}$为1

$$A = (gtmax + 1)\frac{N_{(gtmax+1)}}{N_1} = 4 \times \frac{1}{12} = \frac{1}{3}$$

2-gram	c(a_z)	a_	c(a_)	c(a_z)<gtmax?	c(a_z)+1	n[c(a_z)+1]	n[c(a_z)]	c'(a_z)	折扣率	似然估计值	打折似然估计值
<s> cats	2	<s>	7	TRUE	3	1	4	0.75	0.0625	-0.544068044	-1.748188027
<s> dogs	4	<s>	7	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	-0.243038049	FALSE
<s> the	1	<s>	7	TRUE	2	4	12	0.666666667	0.5	-0.84509804	-1.146128036
bark </s>	1	bark	1	TRUE	2	4	12	0.666666667	0.5	0	-0.301029996
birds </s>	2	birds	3	TRUE	3	1	4	0.75	0.0625	-0.176091259	-1.380211242
birds chirp	1	birds	3	TRUE	2	4	12	0.666666667	0.5	-0.477121255	-0.77815125
cats </s>	2	cats	4	TRUE	3	1	4	0.75	0.0625	-0.301029996	-1.505149978
cats chase	1	cats	4	TRUE	2	4	12	0.666666667	0.5	-0.602059991	-0.903089987
cats meow	1	cats	4	TRUE	2	4	12	0.666666667	0.5	-0.602059991	-0.903089987
chase birds	2	chase	4	TRUE	3	1	4	0.75	0.0625	-0.301029996	-1.505149978
chase cats	1	chase	4	TRUE	2	4	12	0.666666667	0.5	-0.602059991	-0.903089987
chase the	1	chase	4	TRUE	2	4	12	0.666666667	0.5	-0.602059991	-0.903089987
chirp </s>	1	chirp	1	TRUE	2	4	12	0.666666667	0.5	0	-0.301029996
dogs bark	1	dogs	4	TRUE	2	4	12	0.666666667	0.5	-0.602059991	-0.903089987
dogs chase	3	dogs	4	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	-0.124938737	FALSE
meow </s>	1	meow	1	TRUE	2	4	12	0.666666667	0.5	0	-0.301029996
the birds	1	the	2	TRUE	2	4	12	0.666666667	0.5	-0.301029996	-0.602059991
the cats	1	the	2	TRUE	2	4	12	0.666666667	0.5	-0.301029996	-0.602059991

Jelinek-Mercer插值法

- 为了避免出现$P(w)=0$或接近于零的情况，可以用三元、二元和一元相对概率做插值 。

$$\hat{P}(w_t|w_{t-2}w_{t-1}) = \lambda_1 P(w_t|w_{t-2}w_{t-1}) + \lambda_2 P(w_t|w_{t-1}) + \lambda_3 P(w_t)$$

其中$\lambda_1+\lambda_2+\lambda_3=1$。

Kneser-Ney插值法

- 当训练数据非常少的情况下，特别适合采用Kneser-Ney 插值法。Kneser-Ney 是一种非线性插值法，从Absolute discounting（绝对折扣）插值方法演变而来。

- Absolute discounting方法充分利用高阶和低阶语言模型。例如，针对二元语言模型，Absolute discounting平滑公式表示如下：

$$P_{\text{abs}}(w_t|w_{t-1}) = \frac{\max(C(w_{t-1}w_t) - D, 0)}{\sum_{w'} C(w_{t-1}w')} + bP_{\text{abs}}(w_t)$$

其中，$C(w_{t-1}w')$表示$w_{t-1}w'$的组合次数，w'是任意一个词，D是一个固定的折扣值，b是一个规整系数，与折扣值D相关，确保概率分布总和为1。

Kneser-Ney插值法

- 但 $P_{abs}(w_t)$ 可能会存在异常偏大现象。比如“杯子”出现频次较高，因此单独的“杯子”按出现次数可能会比 “茶”多，即P_{abs}(杯子) > P_{abs}(茶)，这样会使Absolute discounting平滑公式因$P_{abs}(w_t)$值过大出现“喝杯子”比“喝茶”概率高的奇怪现象。

Kneser-Ney插值法

- Kneser-Ney插值法对此做了改进，平滑公式如下：

$$P_{\mathrm{KN}}(w_t|w_{t-1}) = \frac{\max(C(w_{t-1}w_t) - D, 0)}{\sum_{w'} C(w_{t-1}w')} + b(w_{t-1})\frac{N_{1+}(w_{t-1}, w_t)}{N_{1+}(w_{j-1}, w_j)}$$

其中，$b(w_{t-1})$是与w_{t-1}相关的规整系数，也可认为是回退权重，即Kneser-Ney平滑融合了插值和回退方法；$w_{j-1}w_j$是任意两个词的组合，$N_{1+}(w_{t-1}, w_t) = |\{w_{t-1}: C(w_{t-1}, w_t) > 0\}|$代表$w_t$之前出现不同词的类数，$N_{1+}(w_{j-1}, w_j) = |\{w_{j-1}: C(w_{j-1}, w_j) > 0\}|$代表所有二元组合的类数。

第一部分的分母可进一步表示为一元模型统计，因此Kneser-Ney 平滑公式还可简化为：

$$P_{\mathrm{KN}}(w_t|w_{t-1}) = \frac{\max(C(w_{t-1}w_t) - D, 0)}{C(w_{t-1})} + b(w_{t-1})\frac{N_{1+}(w_{t-1}, w_t)}{N_{1+}(w_{j-1}, w_j)}$$

语言模型格式(ARPA-MIT)

```
\data\
ngram 1=403172
ngram 2=10149217
ngram 3=5397523

\1-grams:
-2.044557   </s>
-99 <s> -1.478076
-2.402091   一  -1.111727
-4.154127   一一    -0.3032174
-5.838713   一一一  -0.2430122
-6.927923   一一三  -0.1209983
-6.927923   一一五  -0.1209983
-6.927923   一一列举    -0.1209983
-6.927923   一一点破    -0.1209983
-6.717142   一一道来    -0.1209983
-6.927923   一丁    -0.1209983

\2-grams:
-1.808917   <s> 一  -1.036647
-5.008717   <s> 一一    -0.01963902
-6.368144   <s> 一一一
-6.970382   <s> 一一日
-5.2293     <s> 一万    -0.302433
-4.530573   一起 塔式   -0.2380613
-3.96224    一起 填词   -0.1431178
-3.591085   一起 处理   -0.3307952
-3.906852   一起 处罚   -0.1434591
-3.490542   一起 备战   -0.1390026
-3.947429   一起 复习   -0.1419687
-4.397985   一起 复仇
-4.436794   一起 复制
-4.388712   一起 外人
-2.884106   一起 外出   -0.2790813

\3-grams:
-1.735332   凝聚 力量 鼓舞
-0.6466949  分 力量 </s>
-0.1815135  刚性 力量 覆盖
-0.7719197  创新 力量 应该
-0.3742246  到位 力量 到位
-0.2357234  制度 力量 </s>
-0.5506935  制衡 力量 战争
-1.162906   办案 力量 </s>
-0.7141963  办案 力量 中央
-0.8021286  办案 力量 在
-0.818543   办案 力量 抓
-0.3447866  勘探 力量 的
```

\data

ngram 1=nr # 一元语言模型

ngram 2=nr # 二元语言模型

ngram 3=nr # 三元语言模型

\1-grams:

pro_1 word1 back_pro1

\2-grams:

pro_2 word1 word2 back_pro2

\3-grams:

pro_3 word1 word2 word3

\end\

计算出现三个词出现的概率

P(word3|word1,word2)

```
if(存在(word1,word2,word3)的三元模型){
      return pro_3(word1,word2,word3) ;
}else if(存在(word1,word2)二元模型){
      return back_pro2(word1,word2)*P(word3|word2) ;
}else{
      return P(word3 | word2);
}

if(存在(word1,word2)的二元模型){
    return pro_2(word1,word2);
}else{
    return back_pro2(word1)*pro_1(word2) ;
}
```

例:

-3.220352拨打 郑州 -0.4072262

-3.012735郑州 局　-0.3083073

则P（“拨打 郑州 局”）= P（“局 | 拨打 郑州”）= back_pro2(拨打,郑州)*P(局|郑州)=ln($e^{-0.4072262}*e^{-3.012735}$)= -3.4199612。

8.4 语言模型的训练

n-gram训练工具:

- SRI LM Toolkit (n-gram, 支持Unix/Windows平台)
- KenLM (n-gram, 支持Unix/OS X平台)

SRILM

- SRILM由SRI实验室开发，诞生于1995年，包括最大似然估计和平滑算法。

- SRILM主要有两个工具：ngram-count和ngram，用来估计语言模型和计算困惑度。

分词和预处理

- 对于尚未分词的文本进行分词（采用分词工具，如斯坦福大学分词器）

- 处理已经分好词的文本
  - 先根据标点符号{？。！}进行分句
  - 去掉奇怪的符号，比如α @等
  - 替换阿拉伯数字0~9 为 零~九
  - 删除空白行
  - 将连续的空格缩减为一个
  - 删去每行开头的空格和tab键

分词和预处理完的句子示例

—— 体验 了 园 中 各项 运动 项目
—— 化 作 了 这 款 经典 手 包 动人 的 设计 细节
—— 回答 了 青瓷 自 播出 以来 获得 的 各界 反馈
—— 对比 并发 表 感慨 说
—— 拜倒 在 美女 石榴 裙 下
—— 查 摆 施工 生产 中 的 不足
—— 验证 那些 传闻 已 久 的 类 地球
一 丁 认为 爱 恨 妒 构成 了 马 晔 的 人物 性格 特征
一 七旬 老人 在 黔 灵山 公园 晨 练 时 不慎 摔 断 了 腿
一万 个 是 金融 产品 证券 化 产品
一万 个 美好 的 未来 抵 不 上 一个 温暖 的 现在
一万 余 榀 空心 板 梁 陆续 进行 吊装
一万元 为 租 牌 费 一万元 为 押金
一万元 也许 并不 是 最佳 的 出 手 时机
一万元 按 最低 贷款 利息
一万 八千 名 观众 集体 高喊 看
一万多 元 手术 费 和 其他 装 在 包 里面 的 东西 已经 找 不 着 了
一万多 元 的 这种 价 位 已经 很 便宜 了

用SRILM训练语言模型

XIAMEN UNIVERSITY

1. 词频统计

ngram-count -text sample.txt -order 3 -write sample.count

该步骤对分词后的语料进行计数，即统计词频数，其中参数-text表示输入文件sample.txt；-order表示生成几元的语言模型，此处为3元；-write表示输出文件sample.count。

2. 模型训练

ngram-count -read sample.count -order 3 -lm sample.lm

生成的语言模型sample.lm为ARPA文件格式。后面没再接参数，默认采用Good-Turing折扣法和Katz回退法。

如果语料太少，则可采用Witten_Bell折扣法，写法如下：

ngram-count -read sample.count -order 3 -lm sample_wb.lm -wbdiscount

针对较多的语料，普遍采用的是modified Kneser-Ney平滑法，写法如下：

ngram-count -read sample.count -order 3 -lm sample_kn.lm -interpolate -kndiscount

最后两个参数表示平滑算法，其中，-interpolate表示插值平滑（去掉就不会用到低阶信息），-kndiscount表示modified Kneser-Ney平滑法。如果语料太少，计算出来的折扣值可能为负，将导致modified Kneser-Ney平滑法无法采用。

3. 测试（困惑度计算）

ngram -ppl testfile.txt -order 3 -lm sample.lm -debug 2 > file.ppl

用SRILM训练语言模型

4. 模型融合

多个语言模型之间也可以插值合并，以改善模型的效果，特别是对于某些语料较少、难以合并训练的场景。插值合并用法如下：

```
ngram -lm ${mainlm} -order 3 -mix-lm ${mixlm} -lambda 0.8 -write-lm ${mergelm}
```

其中，-lm是第一个n-gram模型，-mix-lm是做插值的第二个n-gram模型。-lambda是主模型（-lm对应的模型）的插值比例，其取值范围为0~1，默认值是0.5。

3-gram模型例子

```
\data\
ngram 1=110485
ngram 2=1997917
ngram 3=1130292

\1-grams:
-1.933266    </s>
-99          <s> -0.4520341
-2.375182    一          -0.5861888
-6.134041    一一列举    -0.1098689
……

\2-grams:
-1.427358    <s> 一      -0.4719597
-1.388696    一 一       -0.6863559
-3.069355    新的 决议   -0.1909096
……

\3-grams:
-2.38712     <s> 一 </s>
-2.458638    <s> 好 </s>
-1.922784    可以 根据 不同
-0.5802485   也 采用 了
……
\end\
```

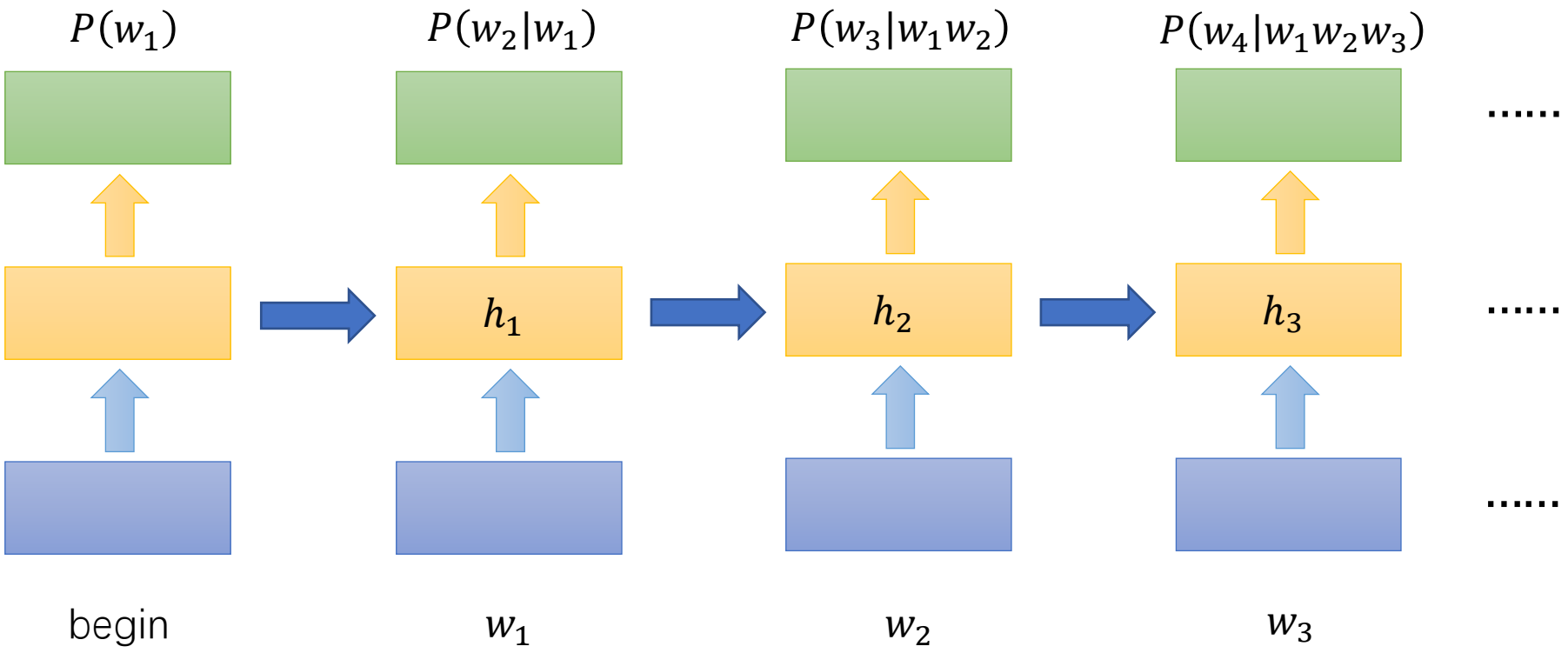
“\data\”下面部分表示1个词、2个词、3个词的组合次数。
“\1-grams:”表示1元模型部分，</s>表示句子结尾，<s>表示句子开头。
在“\3-grams:”表示的3元模型部分，“<s> 好 </s>”表示一句话只出现“好”的对数概率为-2.458638。如果单个字没有类似的组合出现，如“是”没有对应的“<s> 是 </s>”，则单独说“是”，就识别不出来，除非通过回退权重寻找“<s> 是”这样的组合，但概率也会很低。

8.5 神经网络语言模型（NNLM）

- n-gram语言模型一般只能对前3-5个词建模，存在局限性。
- 针对任意长度的句子，我们可采用神经网络（NN），如循环神经网络（RNN），使用循环连接对上下文依赖关系进行建模。

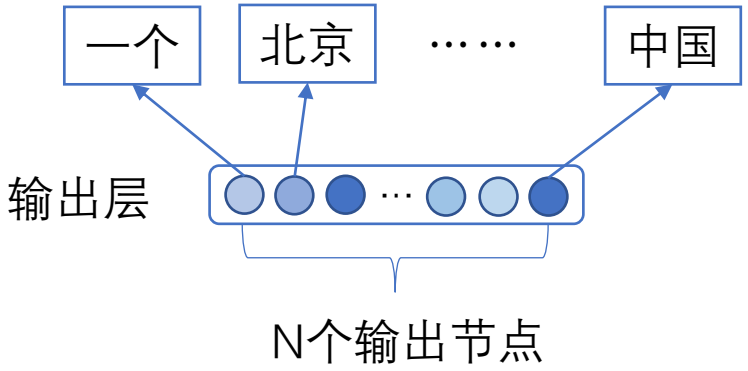
RNNLM

$$P(w_1 \cdots w_{t-1} w_t) = P(w_1)P(w_2|w_1)P(w_3|w_1 w_2) \cdots P(w_t|w_1 w_2 \cdots w_{t-1})$$



RNNLM

词典（N个词）：
一个 ii i2 g e4
北京 b ei3 j ing1
长江 ch ang2 j iang1
黄河 h uang2 h e2
……
中国 zh ong1 g uo2



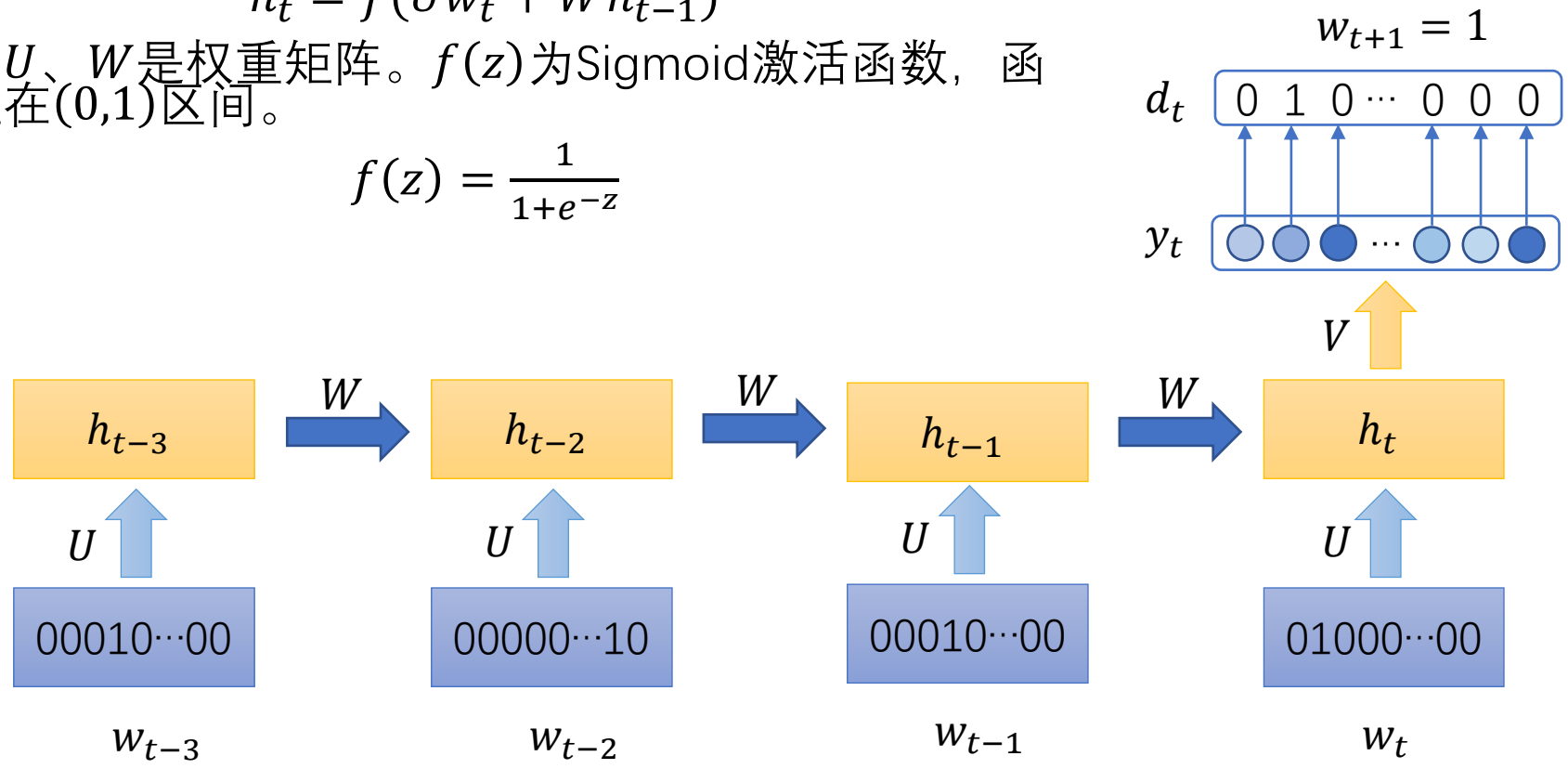
RNNLM

RNN隐藏层的输出h_t由当前词输入w_t和上一个隐藏层输出h_{t-1}联合得到：

$$h_t = f(Uw_t + Wh_{t-1})$$

其中U、W是权重矩阵。$f(z)$为Sigmoid激活函数，函数值在(0,1)区间。

$$f(z) = \frac{1}{1+e^{-z}}$$



词典（N个词）：

0: 一个 ii i2 g e4

1: 北京 b ei3 j ing1

2: 长江 ch ang2 j iang1

3: 黄河 h uang2 h e2

……

N-1: 中国 zh ong1 g uo2

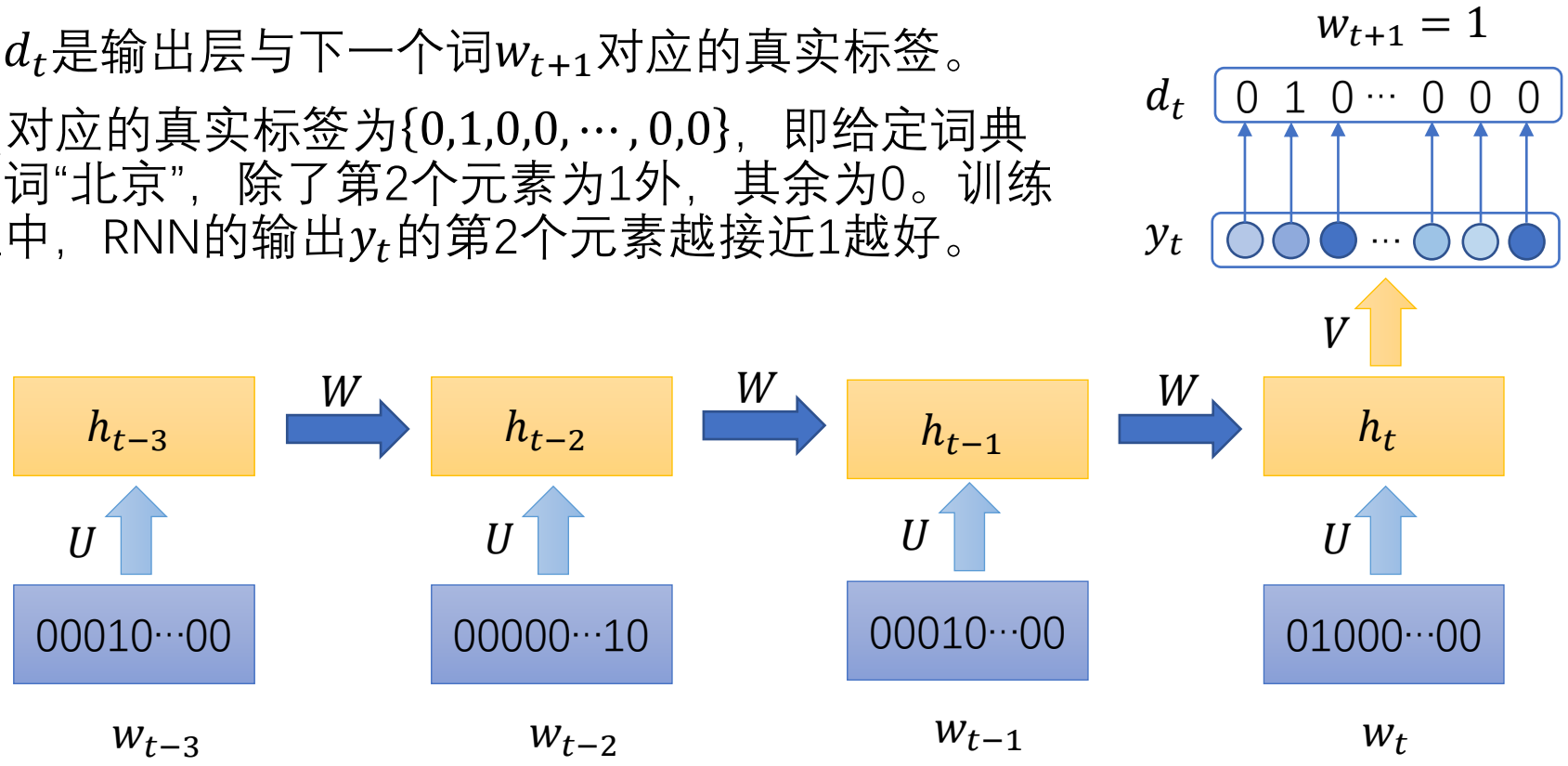
RNNLM错误代价反向传播

错误代价采用交叉熵标准，简化完表示如下：

$$e_t = d_t - y_t$$

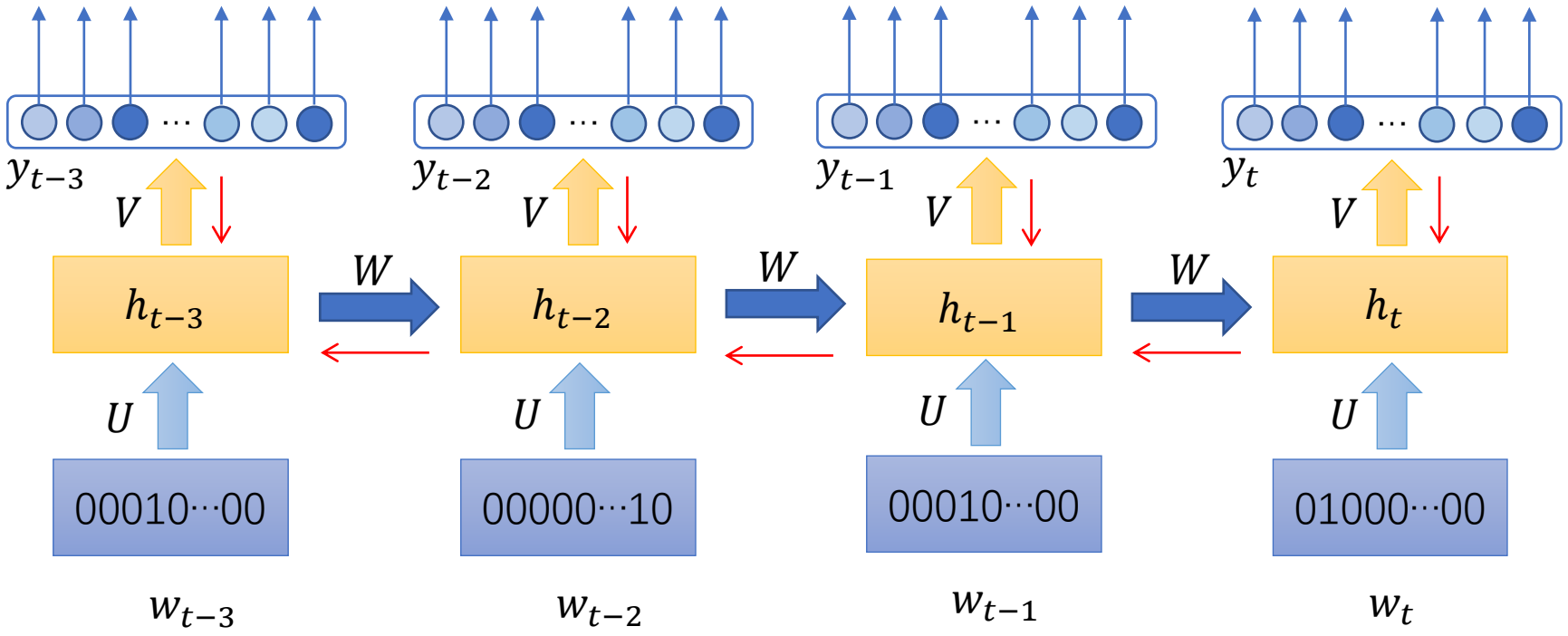
其中d_t是输出层与下一个词w_{t+1}对应的真实标签。

w_{t+1}对应的真实标签为$\{0,1,0,0,\cdots,0,0\}$，即给定词典中的词“北京”，除了第2个元素为1外，其余为0。训练过程中，RNN的输出y_t的第2个元素越接近1越好。



词典（N个词）：
0: 一个 ii i2 g e4
1: 北京 b ei3 j ing1
2: 长江 ch ang2 j iang1
3: 黄河 h uang2 h e2
……
N-1: 中国 zh ong1 g uo2

RNNLM错误代价反向传播

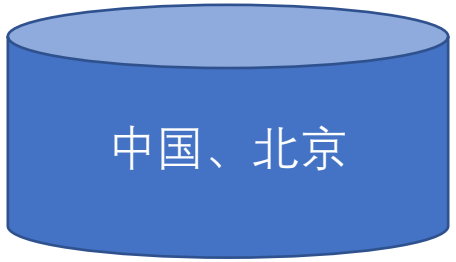


轻量级的NNLM

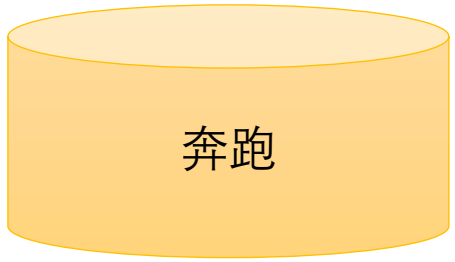
- RNN语言模型相对n-gram有更强大的预测能力，取得更好的识别效果。
- 但由于RNN语言模型输出层的节点数目就是词典的大小，而词典动辄有十几万条甚至更多，计算其概率分布非常复杂，处理大量数据时训练速度将变得异常缓慢，这将极大限制RNN语言模型的使用。
- 为降低运算的复杂度，一种有效办法是对单词进行分类；另一种方案是采用字符建模，减少输出节点数，但声学模型建模单元也要采用字符。

基于类别的语言模型

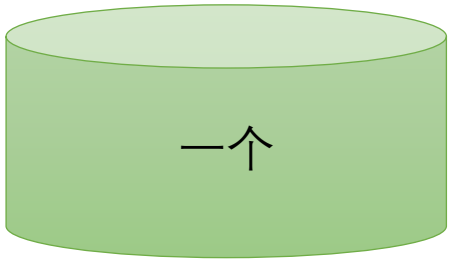
Class1: 名词



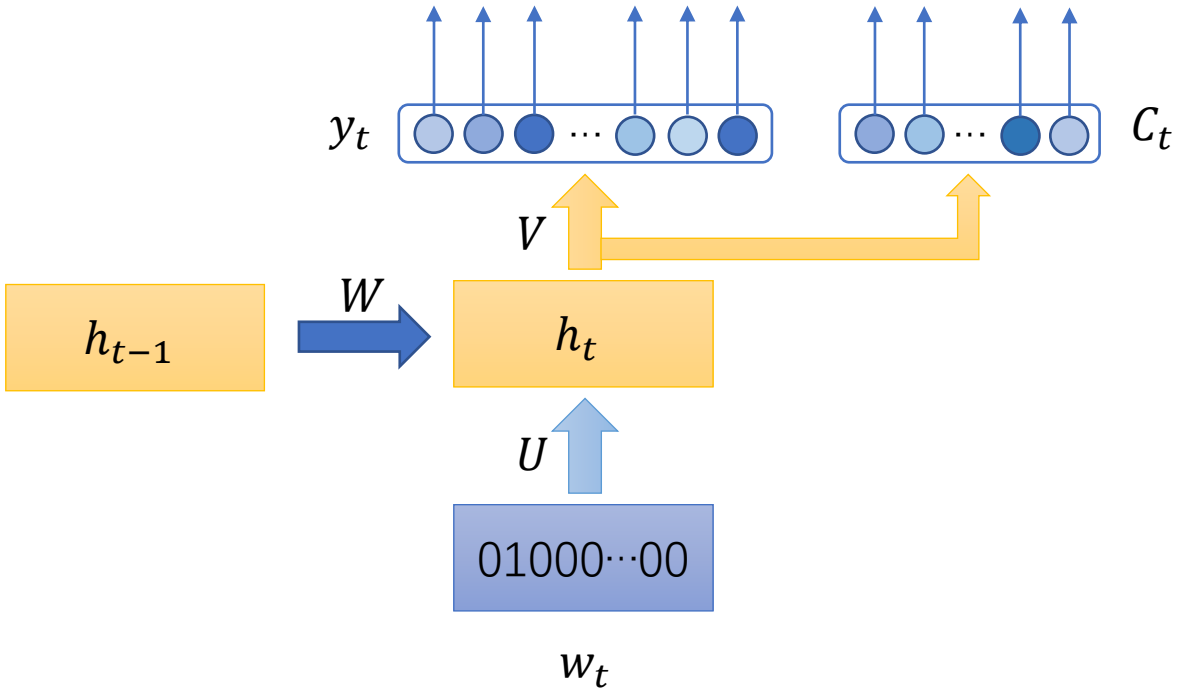
Class2: 动词



Class3: 数量词



类别语言模型的训练



基于类别概率，词与词的组合概率可转化为三部分：

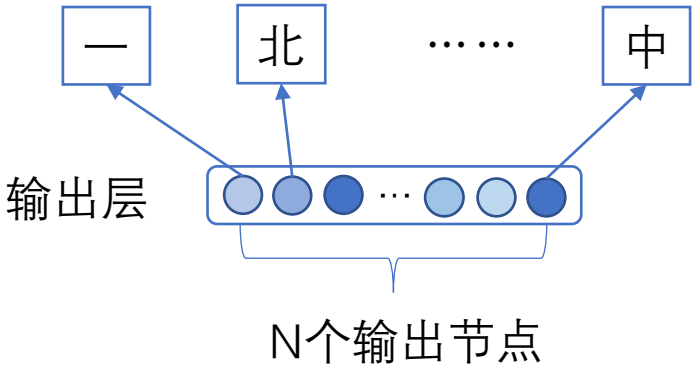
$$P(w_{t+1}|w_t)$$
$$= P(C_{t+1}|C_t)P(w_t|C_t)P(w_{t+1}|C_{t+1})$$

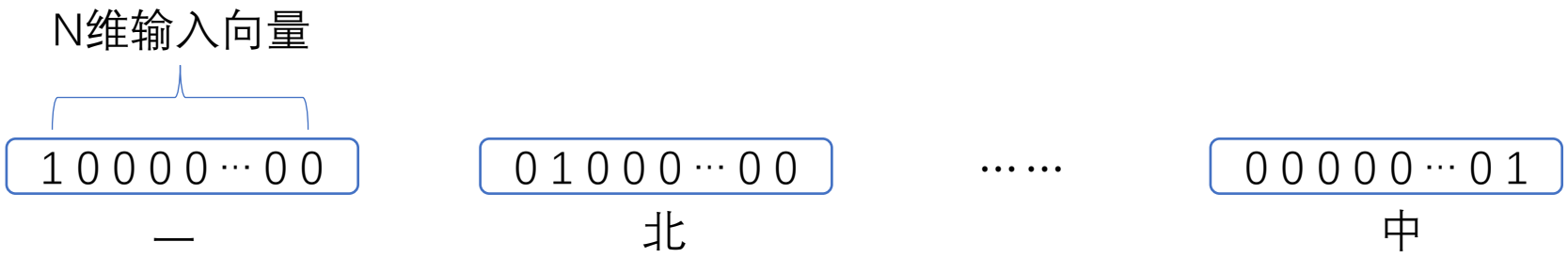
例如：$P(\text{北京}|\text{中国}) = P(\text{名词}|\text{名词})P(\text{中国}|\text{名词})P(\text{北京}|\text{名词})$

其中词与类别的组合概率可通过词频数统计得到。

这样，即使在训练语料里没有“中国 北京”的组合，通过类别间概率和词与类别的概率，也可间接计算出这两个词的组合概率，有效避免训练数据的稀疏。

基于字符的语言模型





可采用Transformer建模。

8.6 本章小结

- 本章详细介绍了统计语言模型，用来预测词序列的组合概率，包括n-gram和NNLM，这两种模型均需要基于大量的文本语料训练得到。

- n-gram预测的词概率值依赖于前$n-1$个词，计算简单，效率高，目前仍是工业界主流的语言模型。而NNLM复杂度高，更多的是用于后处理，在第一次解码之后做二次纠错。