

# 从语言直觉到计算模型

## 汉语自动分词

---

### -N元(统计)语言模型

杨沐昀

哈工大教育部-微软语言语音重点实验室

MOE-MS Joint Key Lab of NLP and Speech (HIT)

# 基于N元语法的分词

- \* 基于N元语法的切分排歧

- \* {Text: 输入文本, 可以是一句话}

$$\hat{Seg} = \arg \max_{Seg} P(Seg | Text)$$

$$= \arg \max_{Seg} \frac{P(Text | Seg) P(Seg)}{P(Text)}$$

$$\propto \arg \max_{Seg} P(Text | Seg) P(Seg)$$

$$= \arg \max_{Seg} P(Seg)$$

对比推导结果：  
为何不直接假设？

?

?

# 基于N元语法的切分排歧

- \* Seg简写为S，含有n个词：  $\{w_1, w_2, \dots, w_n\}$

$$p(S) = p(w_1^n) = p(w_1) \cdot \prod_{i=2}^n p(w_i | w_1^{i-1})$$

- \* MM模型/过程：有限历史假设，仅依赖前n-1个词
  - \* 一种最简化的情况

$$P(S) = p(w_1) \cdot p(w_2) \cdot p(w_3) \dots p(w_n)$$

#连乘的代码实现?

# 基于N元语法的切分排歧

- \* 采用一元语法

- \* 等价于最大频率分词
- \* 即把切分路径上每一个词的词频相乘得到该切分路径的概率
- \* 把词频的负对数理解成“代价”，这种方法也可以理解为最少分词法的一种扩充
- \* 正确率可达到92%
- \* 简便易行，效果一般好于基于词表的方法

# 基于N元语法的切分排歧

- ❖ 采用二元语法：性能可以进一步提高

$$p(S) = p(w_1) \cdot p(w_2 | w_1) \cdot p(w_3 | w_2) \cdots p(w_n | w_{n-1})$$

- ❖ 采用更大的N:利用更多上下文信息

- ❖ 参数空间

词表=20,000

$n$	n-gram的个数
2 (bigrams)	400,000,000
3 (trigrams)	8,000,000,000,000
4 (4-grams)	$1.6 \times 10^{17}$

#解决参数爆炸的策略？得失？

# 语言模型：等价类映射

- \* 绝大多数历史不会出现在训练数据中。
- \* 将历史  $\omega_1 \omega_2 \dots \omega_{i-1}$  映射到等价类  $E(\omega_1 \omega_2 \dots \omega_{i-1})$ ，其中等价类的数目远小于全部历史的数目。
- \* 假设：  $p(\omega_i | \omega_1 \dots \omega_{i-1}) = p(\omega_i | E(\omega_1 \omega_2 \dots \omega_{i-1}))$ ，则自由参数的数目会大大减少。
- \* 可靠性与辨别力
  - \* 更大的n：对下一个词出现的约束性信息更多，更大的辨别力。
  - \* 更小的n：在训练语料库中出现的次数更多，更可靠的统计结果，更高的可靠性。

# 基于N元语法的切分排歧

## \* 基于HMM的分词词性标注一体化模型

Given Sentence, Output Seg & POS

$P(\text{POS} \mid \text{Sentence})$

$\propto P(\text{Sent} \mid \text{POS}) P(\text{POS})$

$\Rightarrow P(W \mid \text{POS}) \cdot P(\text{POS})$

\*Sentence=W (分词序列)

## \* 采用二元文法计算上式

$$P(w_1 \mid \text{POS}_1) \cdot P(\text{POS}_1 \mid S_{\text{begin}}) \cdot P(w_2 \mid \text{POS}_2) \cdot P(\text{POS}_2 \mid \text{POS}_1) \cdot \dots \cdot P(w_n \mid \text{POS}_n) P(\text{POS}_n \mid \text{POS}_{n-1})$$

# 基于N元语法的切分排歧

- \* 我们先考虑模型的初步应用
  - \* 给定大规模标记语料库
  - \* 完成HMM的参数估计：MLE
  - \* 一个具体的分词+词性标注是如何完成的？
  - \* （课后补充阅读：平滑算法）

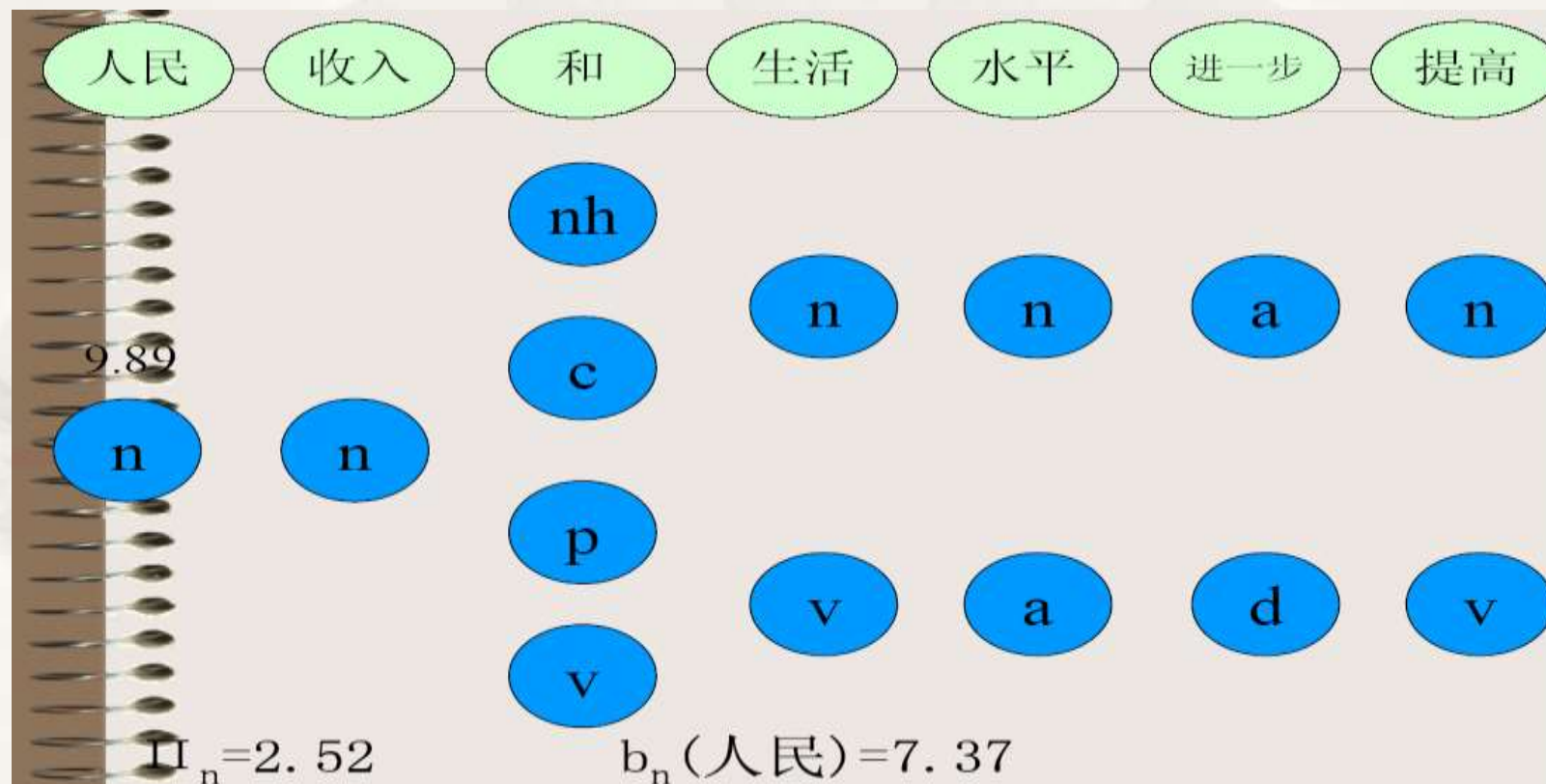


# 基于统计的词网格分词

- \* 第一步是候选词网格构造：利用词典匹配，列举输入句子所有可能的切分词语，并以词网格形式保存
- \* 第二步计算词网格中的每一条路径的权值
- \* 根据图搜索算法在图中找到一条权值最大的路径，作为最后的分词结果
- \* 动态规划算法：Viterbi算法
  - \* 曾有A\*启发式搜索算法的实现 (necessary?)

# Viterbi搜索——例子

分词“词图”中的某段局部路径：



本例出处待考，谨在此致谢！

人民 收入 和 生活 水平 进一步 提高

nh

n

n

a

n

c

p

v

v

a

d

v

9.89

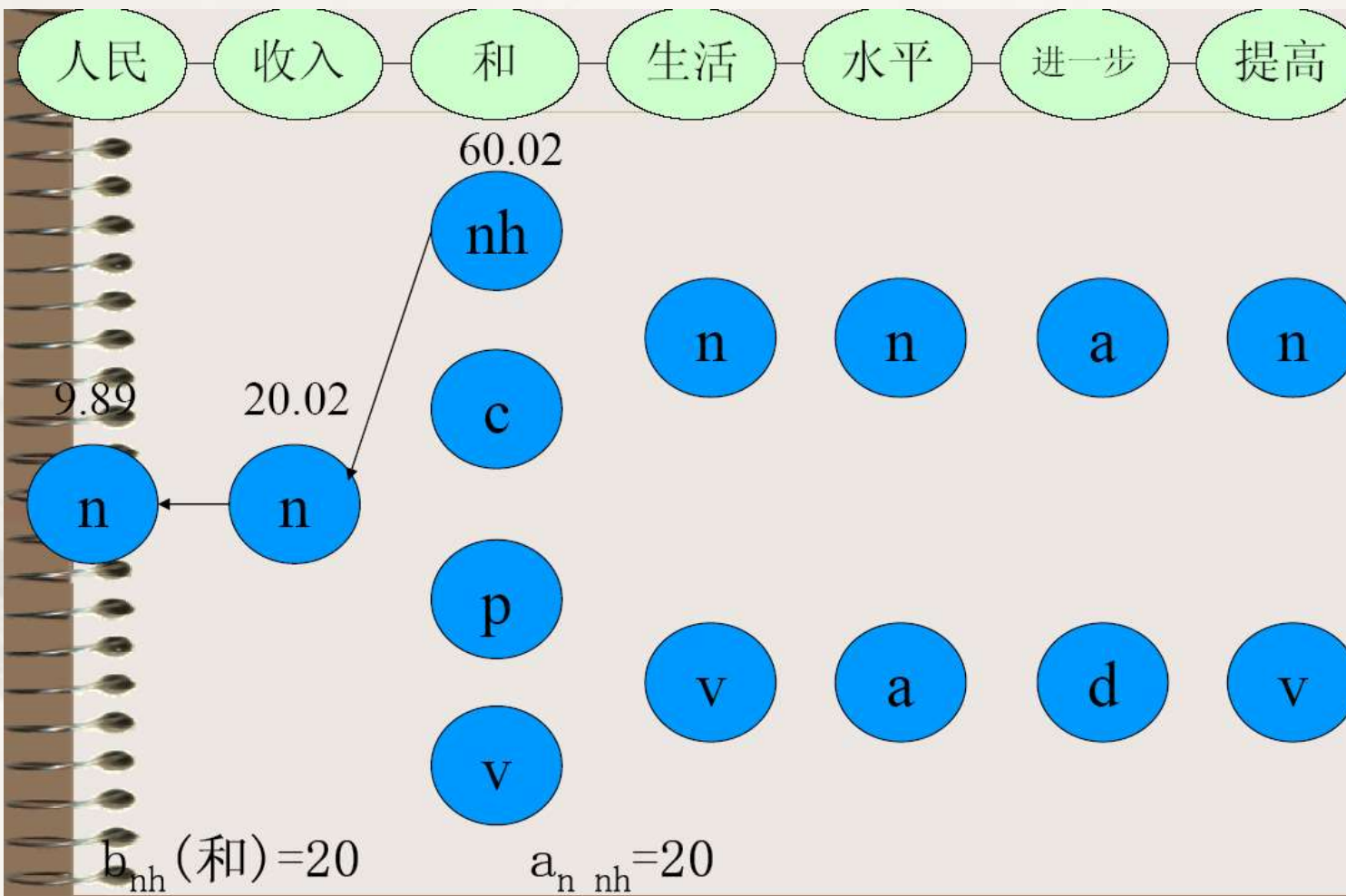
20.02

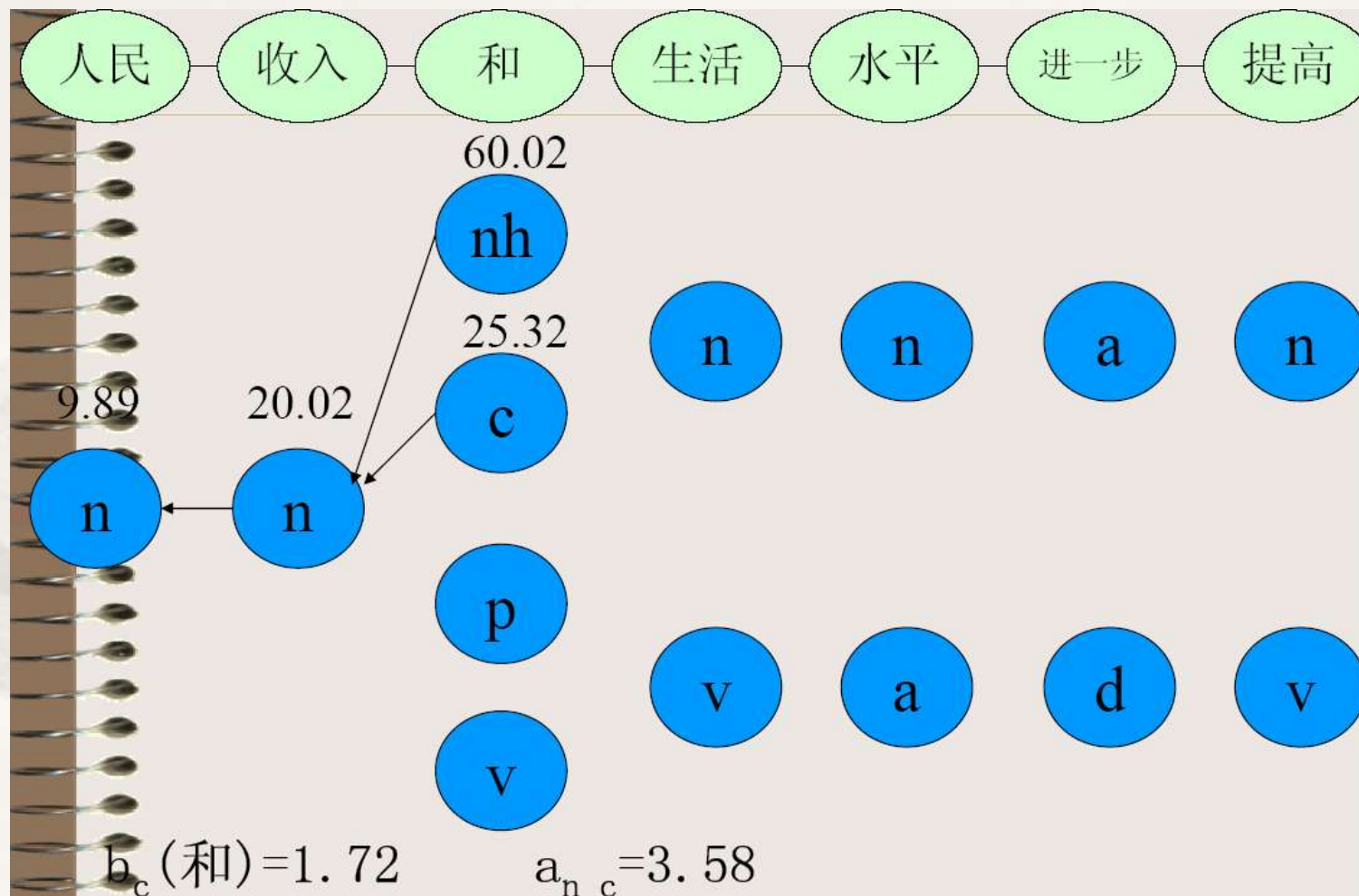
n

n

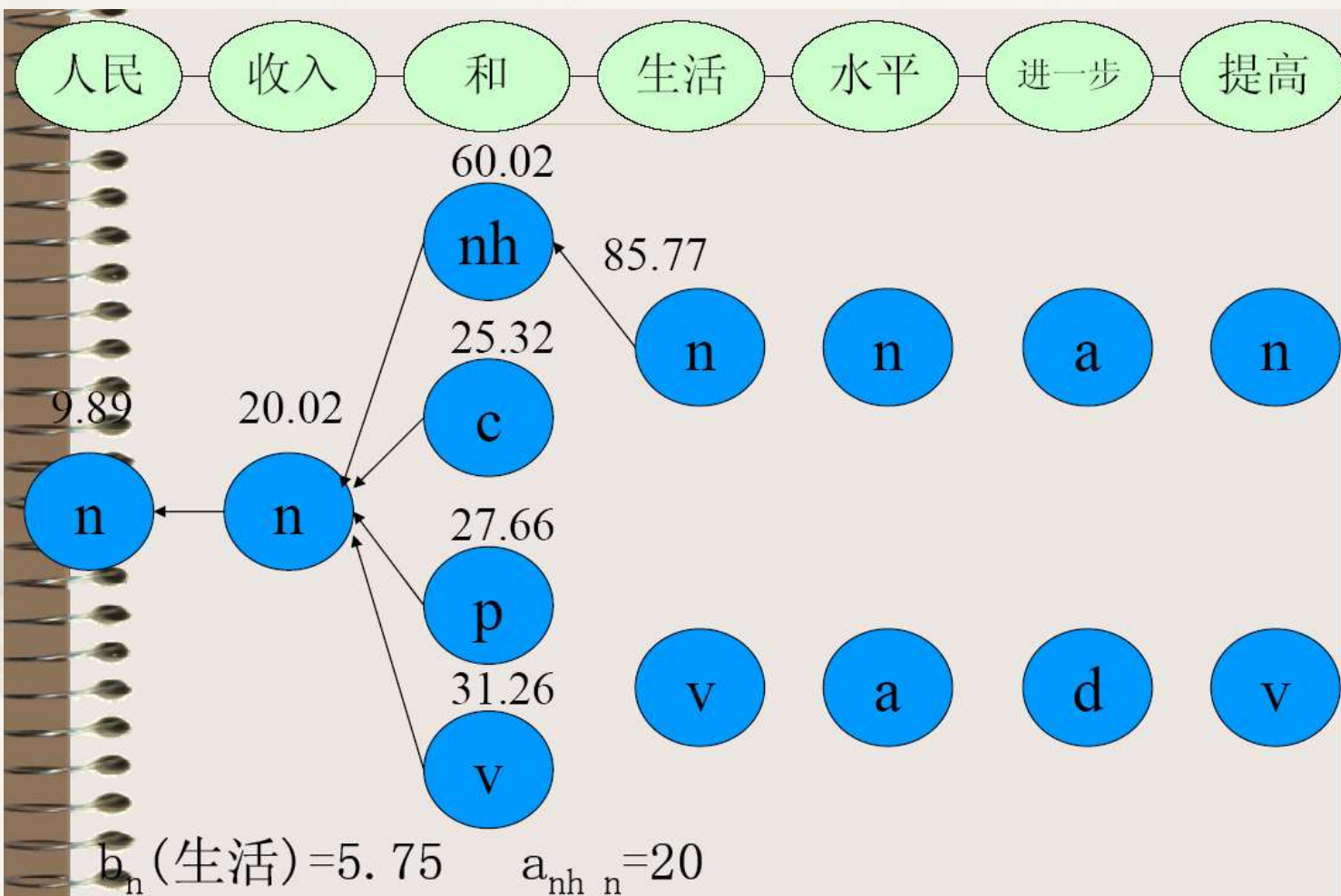
$b_n(\text{收入}) = 6.98$

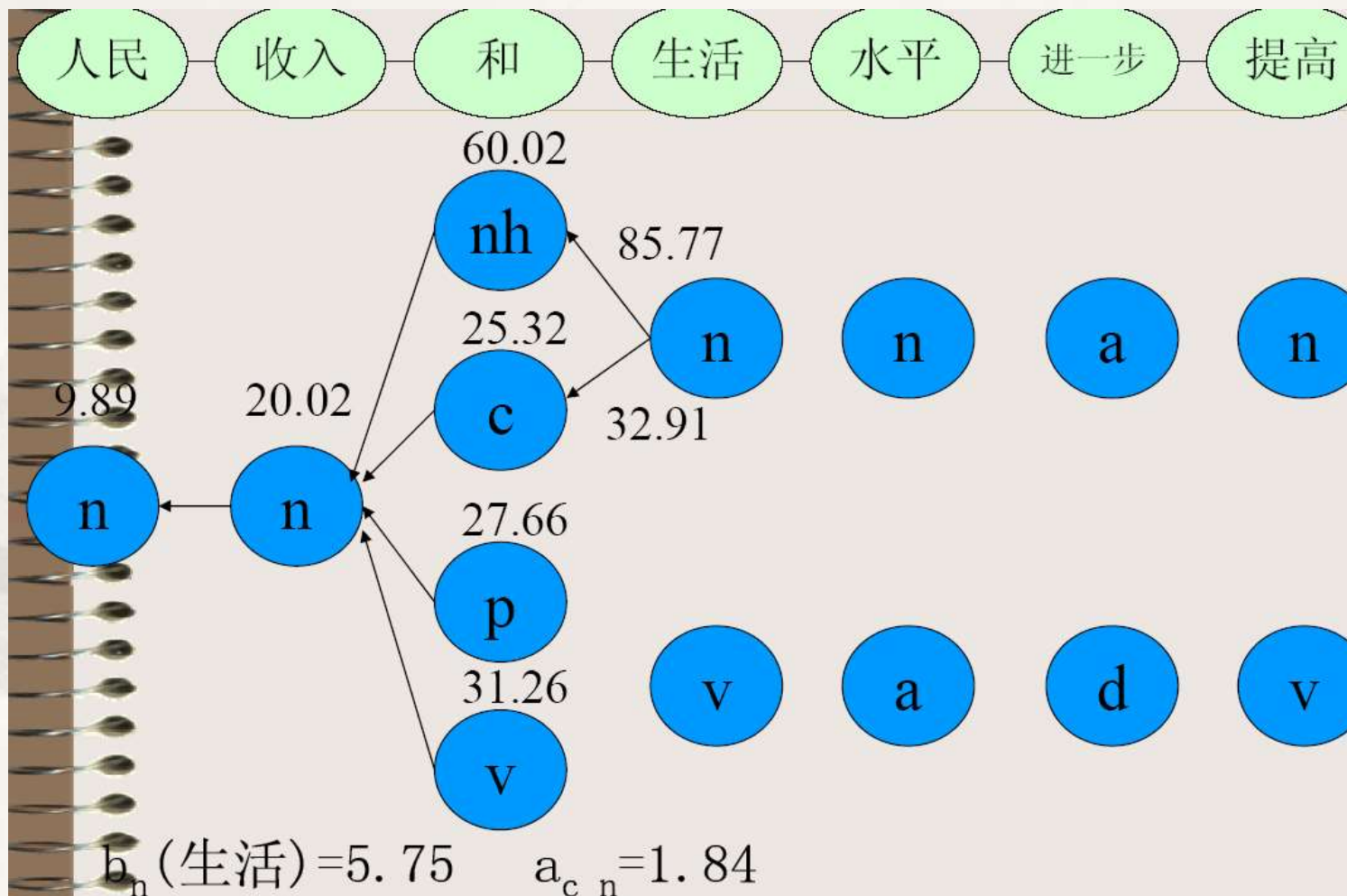
$a_{nn} = 3.15$

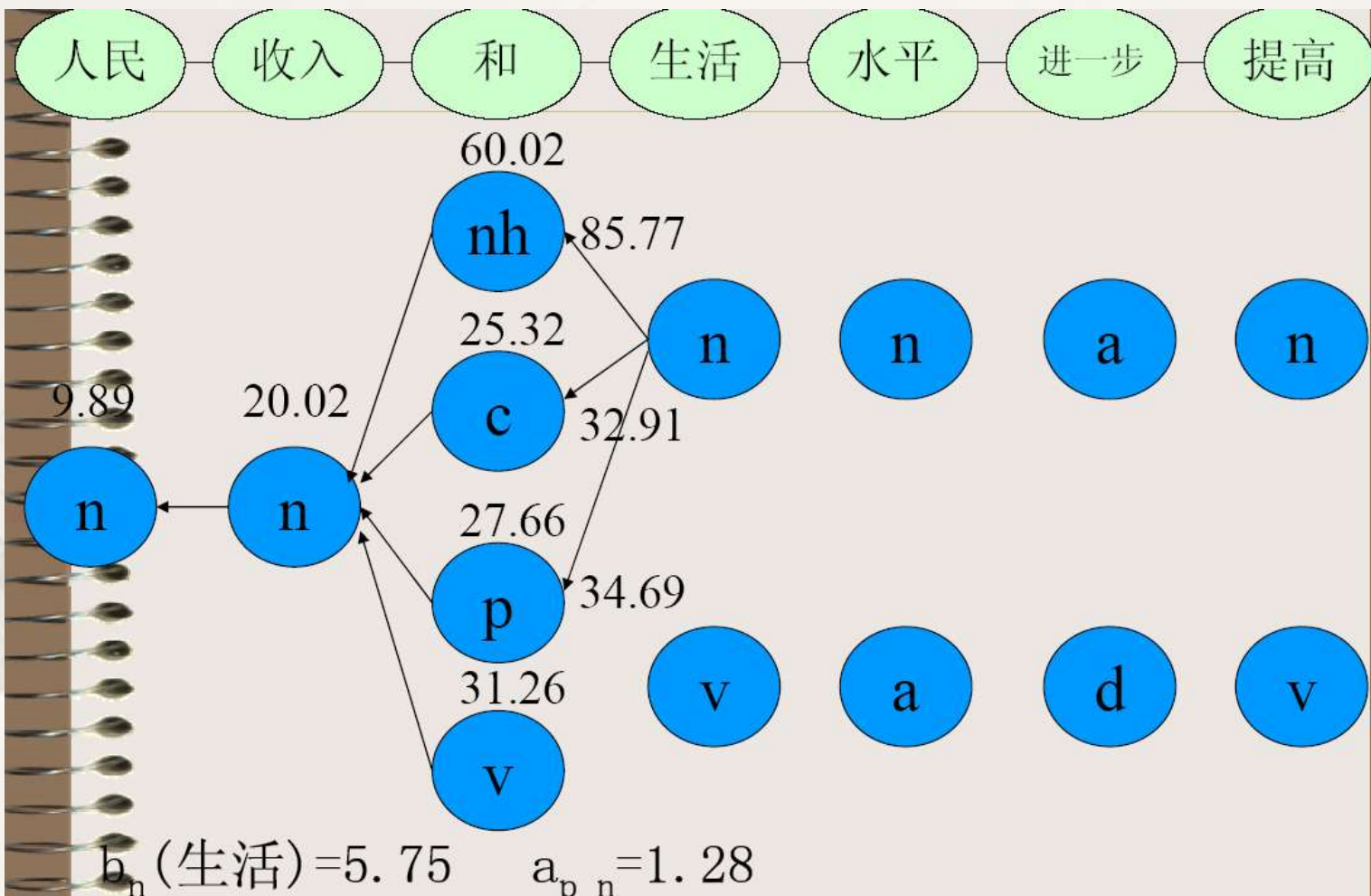




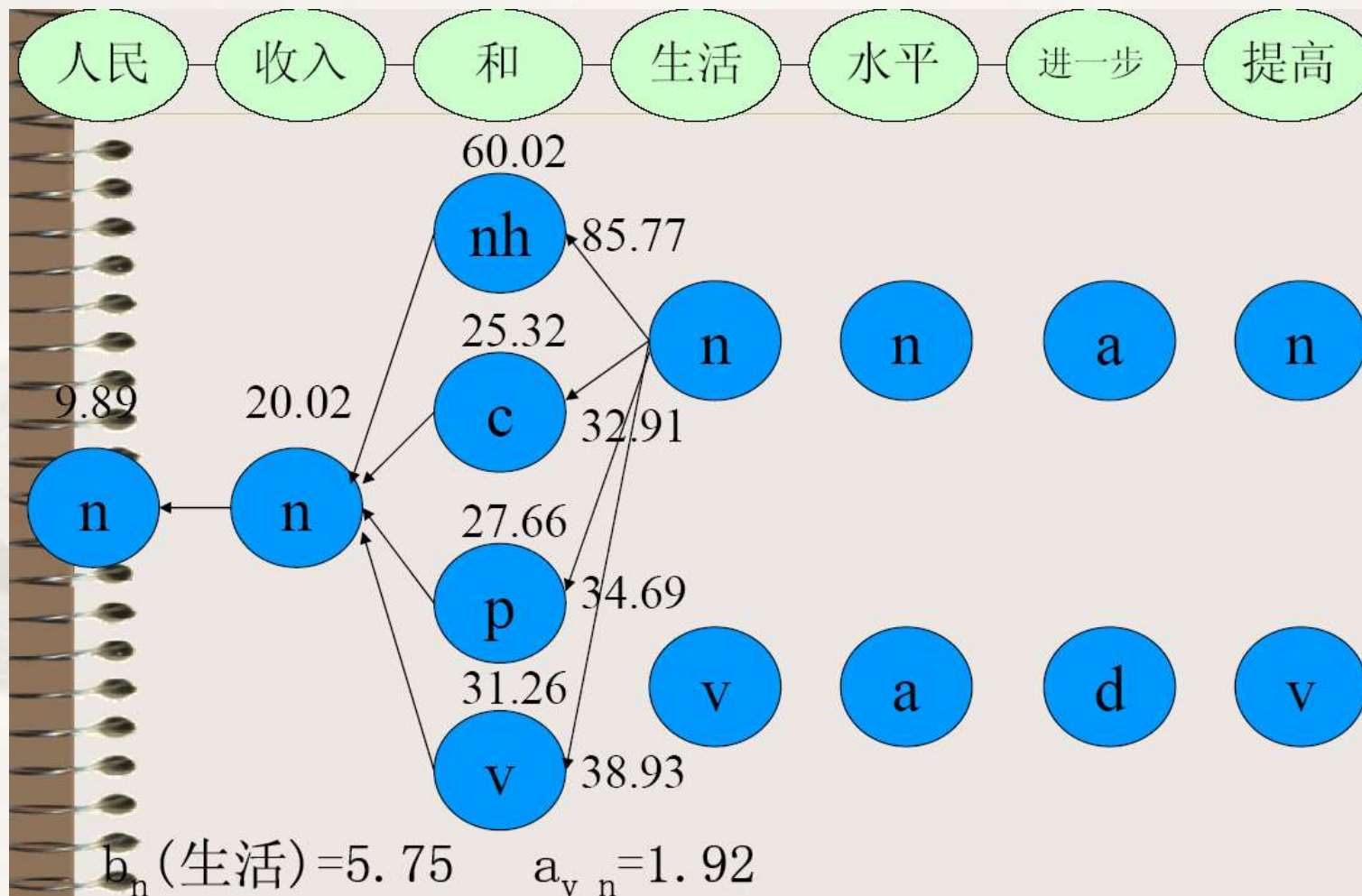




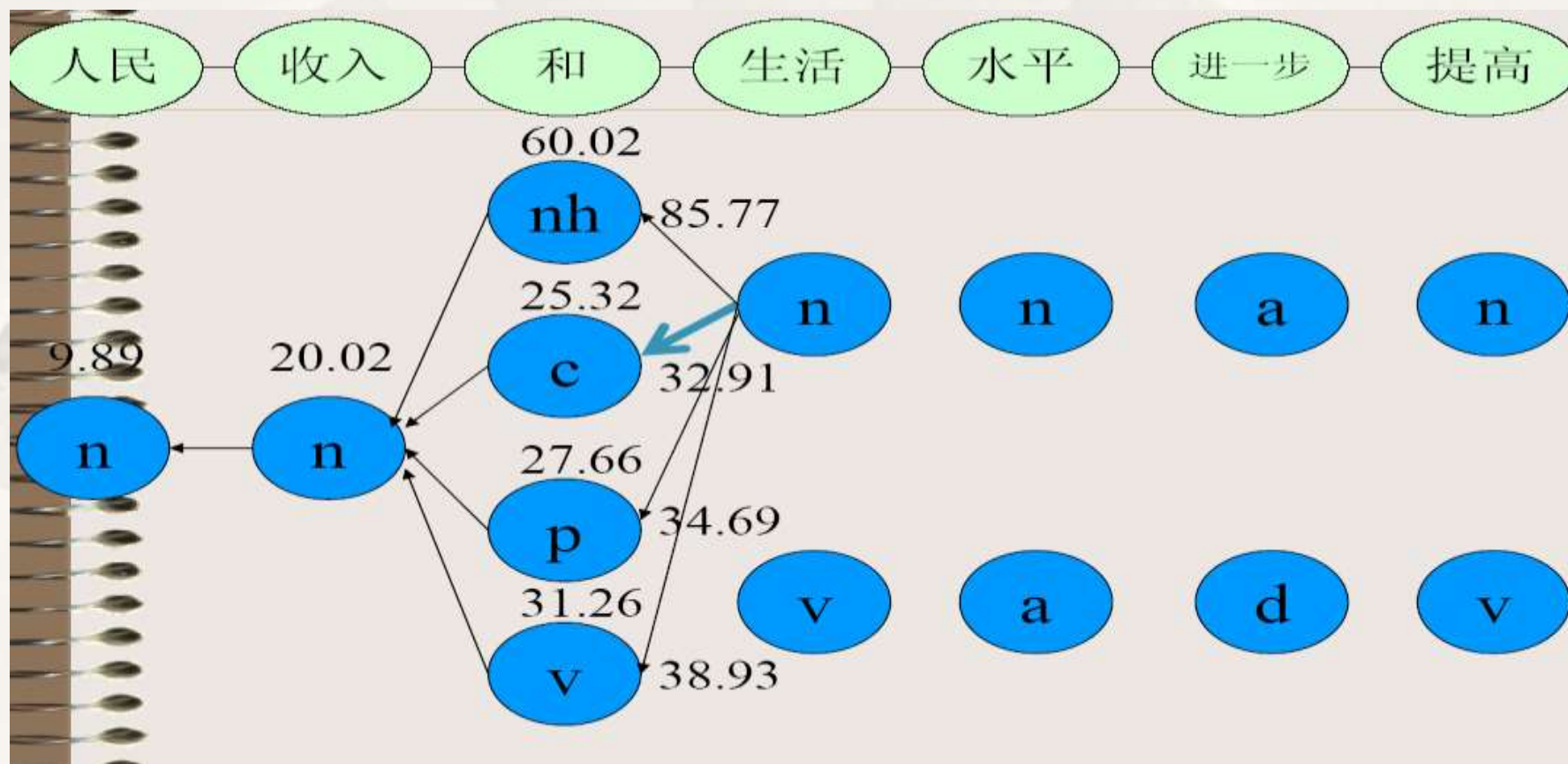


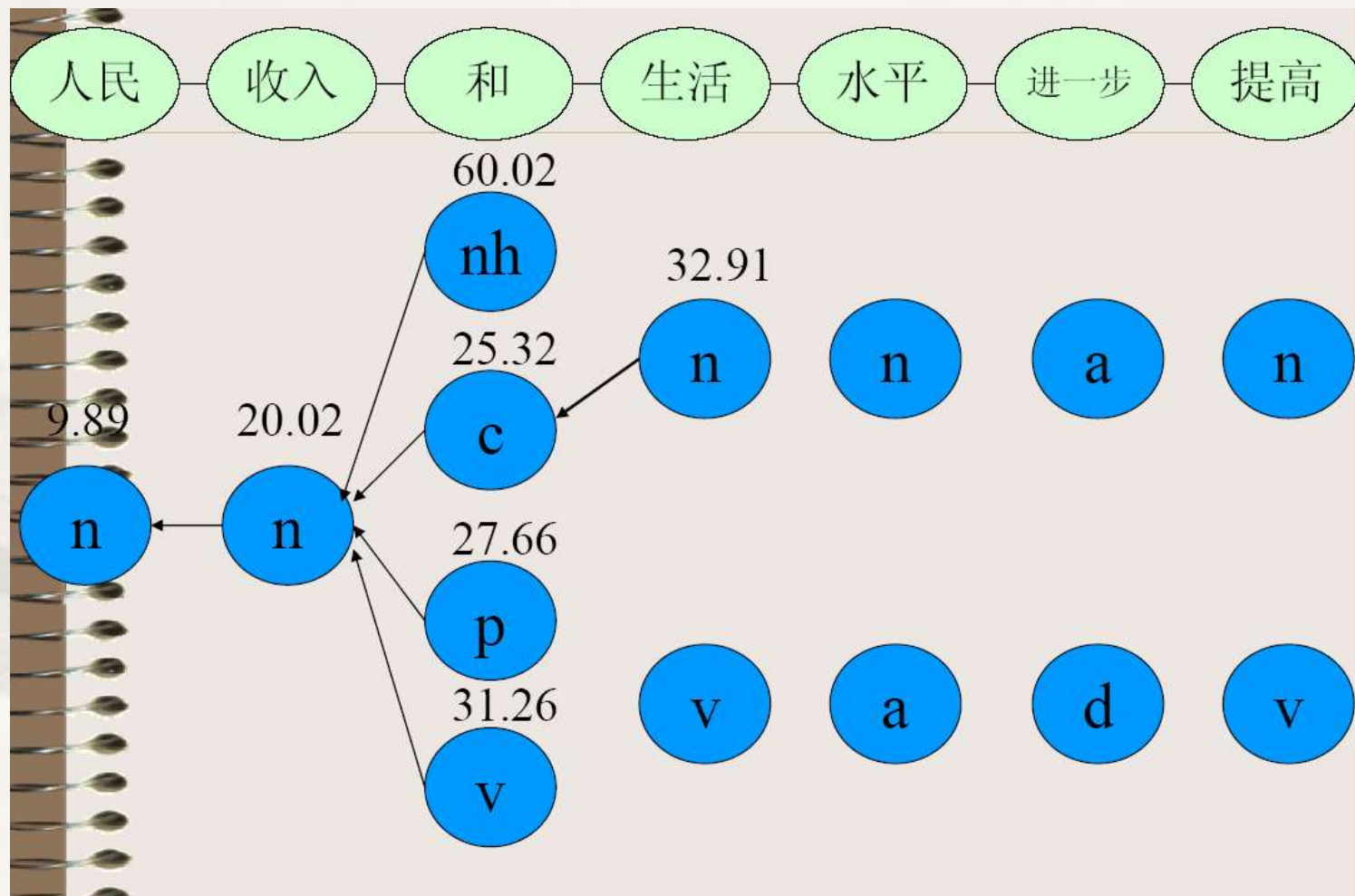


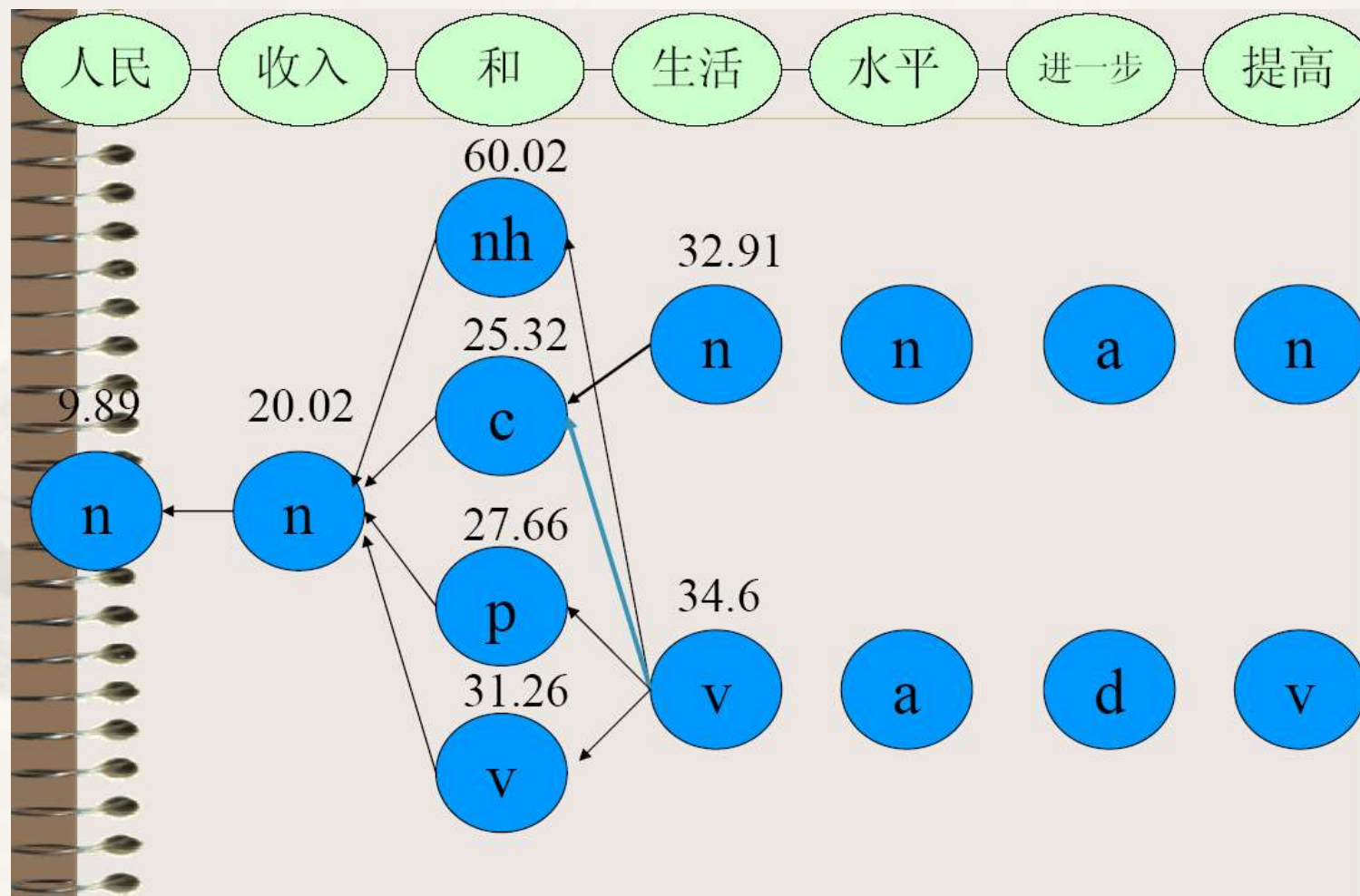




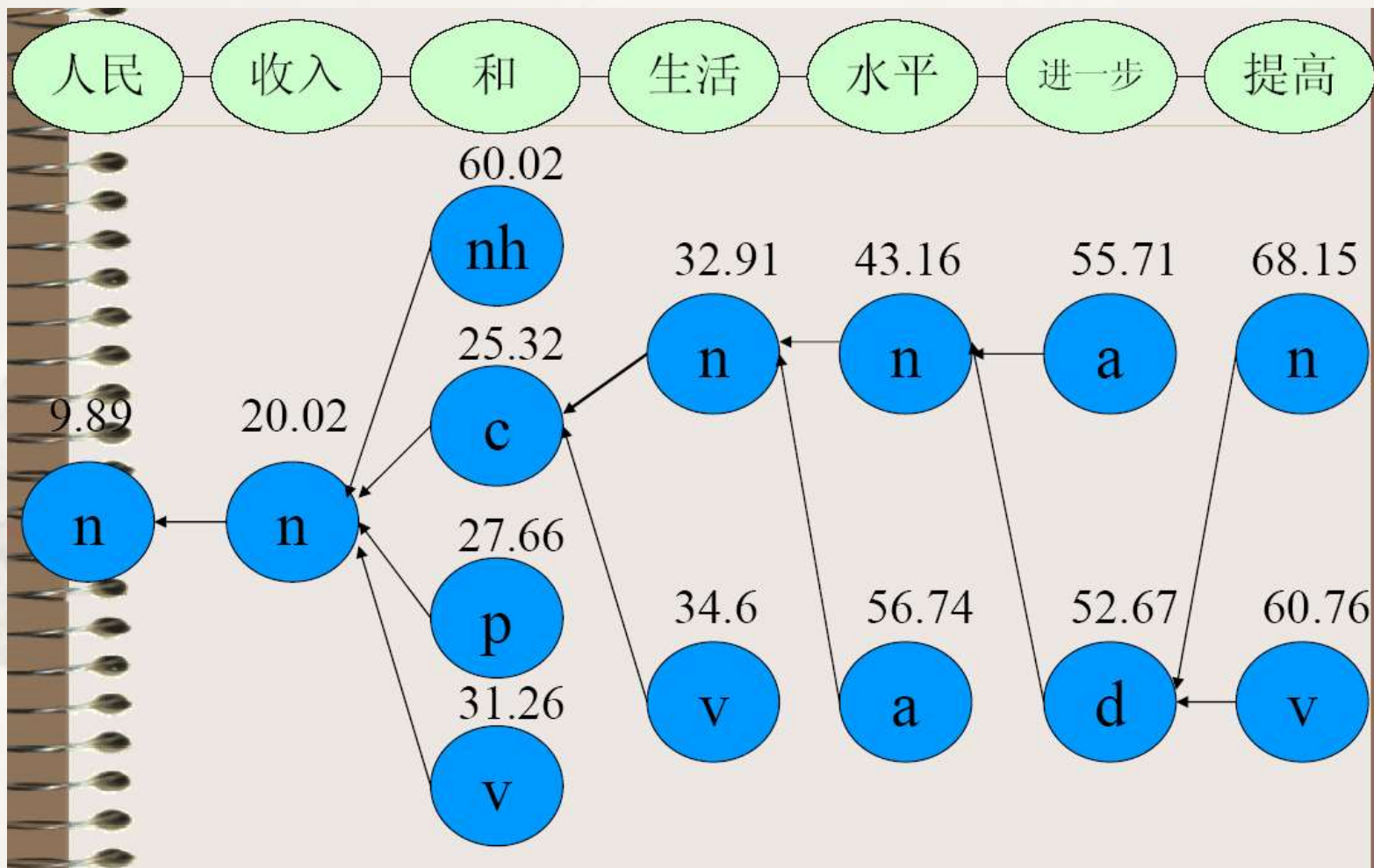
记录“和→生活”的最佳路径是“c-n”



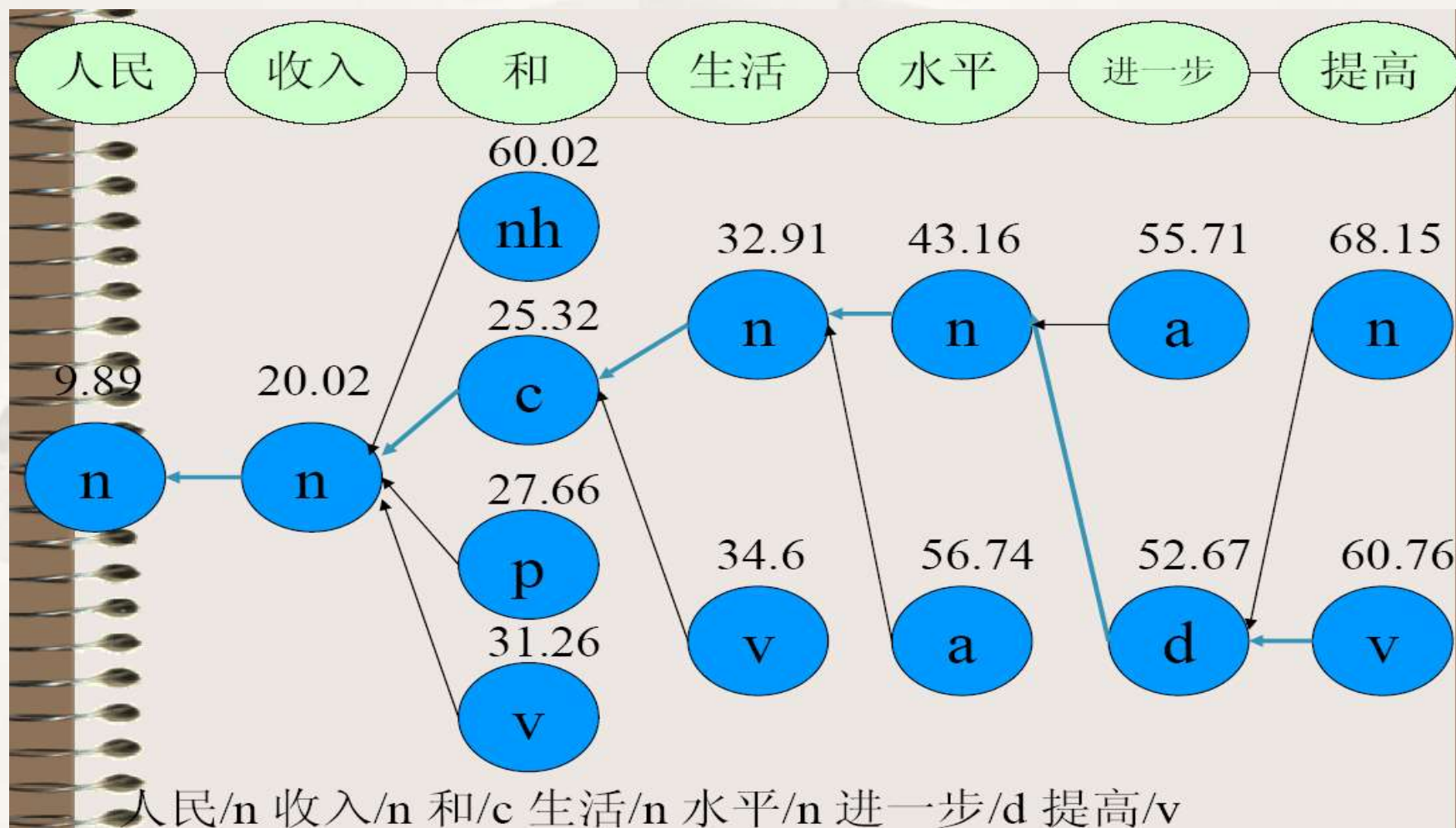








提取所有的“最优路径”记录，确定最优标记状态



# 中文未登录词识别

---

# 未登录词的类型

- \* 命名实体（Named Entity）

- \* 汉语人名：李素丽 老张 李四 王二麻子
- \* 汉语地名：定福庄 白沟 三义庙 韩村 河马甸
- \* 翻译人名：乔治·布什 叶利钦 包法利夫人
- \* 翻译地名：阿尔卑斯山 新奥尔良 约克郡
- \* 机构名：方正公司 联想集团 国际卫生组织外贸部

- \* 数字、日期词、货币等

- \* 商标字号：非常可乐 乐凯 波导 杉杉 同仁堂

- \* 专业术语：万维网 主机板 模态逻辑 贝叶斯算法

- \* 缩略语：三个代表 五讲四美 打假扫黄 打非计生办

- \* 新词语：卡拉OK 波波族 美刀 港刀



# 未登录词识别的依据

---

- \* 内部构成规律（用字规律）
- \* 外部环境（上下文）
- \* 重复出现规律

# 中国人名的内部构成规律

- \* 在汉语的未定义词中，中国人名是规律性最强，也是最容易识别的一类；
- \* 中国人名一般由以下部分组合而成：
  - \* 姓：张、王、李、刘、诸葛、西门、范徐丽泰
  - \* 名：李素丽，张华平，王杰、诸葛亮
  - \* 前缀：老王，小李
  - \* 后缀：王老，赵总
- \* 中国人名各组成部分用字比较有规律

# 中国人名的内部构成规律

- \* 台湾出版的《中国姓氏集》收集姓氏5544个，其中，单姓3410个，复姓1990个，3字姓144个。
- \* 中国目前仍使用的姓氏共737个，其中，单姓729个，复姓8个。
- \* 根据我们收集的300万个人名统计：姓氏：974个，其中，单姓952个，复姓23个，300万人名中出现汉字4064个。

# 中国人名的内部构成规律

- \* 中国人名各组成部分的组合规律
  - \* 姓 + 名
  - \* 姓
  - \* 名
  - \* 前缀 + 姓
  - \* 姓 + 后缀
  - \* 姓 + 姓 + 名（海外已婚妇女）

# 中国人名的上下文构成规律

## \* 身份词:

- \* 前: 工人、教师、影星、犯人
- \* 后: 先生、同志
- \* 前后: 女士、教授、经理、小姐、总理

## \* 地名或机构名:

- \* 前: 静海县大丘庄禹作敏

## \* 的字结构

- \* 前: 年过七旬的王贵芝

## \* 动作词

- \* 前: 批评, 逮捕, 选举
- \* 后: 说, 表示, 吃, 结婚

# 中国人名识别的难点

- \* 一些高频姓名用字在非姓名中也是高频字
  - \* 姓氏：于，马，黄，张，向，常，高
  - \* 名字：周鹏和同学，周鹏和同学
- \* 人名内部相互成词，指姓与名、名与名之间本身就是一个已经被收录的词
  - \* [王国]维、[高峰]、[汪洋]、张[朝阳]
- \* 人名与其上下文组合成词
  - \* 这里[有关]天培的壮烈；
  - \* 费孝通向人大常委会提交书面报告
- \* 人名地名冲突: 河北省刘庄

# 中文姓名识别方法

## \* 中文姓名识别方法

- \* 姓名库匹配，以姓作为触发信息，寻找潜在的名字
- \* 计算潜在姓名的概率估值及相应姓氏的姓名阈值，根据姓名概率评价函数和修饰规则对潜在的姓名进行筛选。

# 中国地名的识别

## \* 困难

- \* 地名数量大，缺乏明确、规范的定义。  
《中华人民共和国地名录》（1994）收集88026个，不包括相当一部分街道、胡同、村庄等小地方名称。
- \* 真实语料中地名出现情况复杂。如地名简称、地名用词与其它普通词冲突、地名是其它专用名词的一部分，地名长度不一等。



# 未登录词识别的一般方法

- \* 在统计方法中，未登录词识别的一种最通常的做法就是将识别问题转化成标注问题
- \* 对于输入句子中的每个汉字，定义四个标记：
  - \* 不属于未登录词O
  - \* 未登录词首字B
  - \* 未登录词尾字E
  - \* 未登录词中间字I


# 将识别问题转化成标注问题

- \* 如果能够把输入句子中的每个汉字都正确地按上述标记进行标注，那么未登录词的识别自然就解决了
- \* 标注可以采用
  - \* 隐马尔科夫模型 (HMM)
  - \* 最大熵 (ME)
  - \* 最大熵马尔科夫模型 (MEMM)
  - \* 条件随机场 (CRF) 等

# 基于HMM的未登录词识别

- \* 以人名识别为例，输入文本：
  - \* 这是周恩来、邓颖超生前居住的地方
  - \* 标注为：
  - \* 这是周恩来、邓颖超生前居住的地方
  - \* O O B I E O B I E O O O O O O O
  - \* 两处标注为BIE的字串“周恩来”、“邓颖超”被识别为人名
- \* 训练语料库为已经标注人名的语料库

---



Q & A!

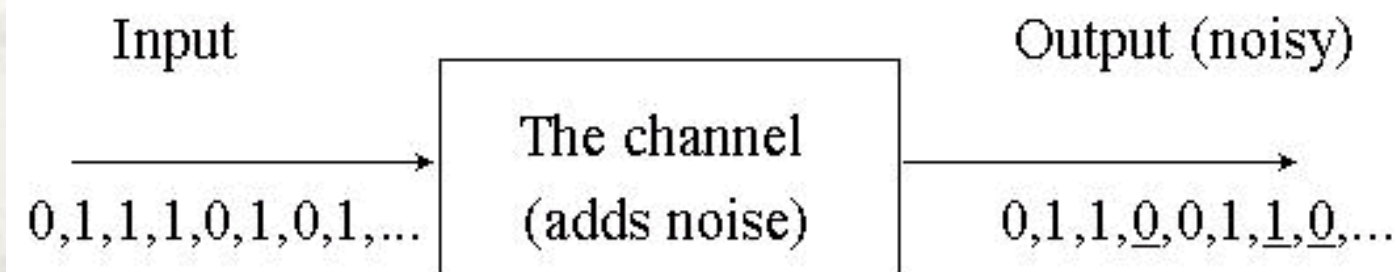
# 课下阅读

---

- \* N元语言模型
  - \* 一些基本概念
  - \* 模型评价

# 语言模型：噪声信道模型

- \* 噪声信道模型



- \* 模型：出错的概率。
- \* 举例：  $p(0|1)=0.3$ ,  $p(1|1)=0.7$ ,  $p(1|0)=0.4$ ,  $p(0|0)=0.6$
- \* 任务是：
  - \* 已知带有噪声的输出，想知道输入是什么。

# 语言模型：香农游戏

---

- \* Claude E. Shannon. “Prediction and Entropy of Printed English”, *Bell System Technical Journal* 30:50–64. 1951.
- \* 给定前 $n-1$ 个词(或者字母), 预测下一个词(字母)
- \* 从训练语料库中确定不同词序列概率

# n元语法(n-gram)：基本概念

- \* 马尔科夫假设：下一个词的出现仅依赖它前面的一个词或几个词。
- \* 将两个历史 $\omega_{i-n+2} \dots \omega_{i-1} \omega_i$ 和 $v_{k-n+2} \dots v_{k-1} v_k$ 映射到同一个等价类，当且仅当这两个历史最近的 $n-1$  ( $1 \leq n \leq l$ ) 个词相同。
- \* 即若 $E(\omega_1 \omega_2 \dots \omega_{i-1} \omega_i) = E(v_1 v_2 \dots v_{i-1} v_i)$ ，则 $(\omega_{i-n+2} \dots \omega_{i-1} \omega_i) = (v_{k-n+2} \dots v_{k-1} v_k)$
- \* 满足上述条件的语言模型称为n元语法或n元文法。



# n元语法(n-gram)：基本概念

- \* 一元文法：n=1时，出现在第 $i$ 位上的词 $\omega_i$ 独立于历史，记作unigram。
- \* 二元文法：n=2时，出现在第 $i$ 位上的词 $\omega_i$ 只与前面的一个历史词 $\omega_{i-1}$ 有关，记作bigram，也被称为一阶马尔科夫链。
- \* 三元文法：n=3时，出现在第 $i$ 位上的词 $\omega_i$ 只与与前面的两个历史词 $\omega_{i-1}\omega_{i-2}$ 有关，记作trigram，也被称作二阶马尔科夫链。

# 模型评价

---

- \* 实用方法：
  - \* 通过查看该模型在实际应用中的表现来评价统计语言模型。
    - \* 优点：直观，实用
    - \* 缺点：缺乏针对性，不够客观
- \* 理论方法：
  - \* 交叉熵与困惑度（也称迷惑度，perplexity）

# 模型评价：熵

- \* 如果 $X$ 是一个离散型随机变量，取值空间为 $R$ ，其概率分布为：

$$p(x) = P(X = x), x \in R$$

那么， $X$ 的熵 $H(x)$ 定义为：

$$H(X) = - \sum_{x \in R} p(x) \log_2(x)$$

其中，约定 $0 \log 0 = 0$ 。

- \* 熵又称为自信息(self-information)，可以视为描述一个随机变量的不确定性的数量，它表示信源 $X$ 每发一个符号所提供的平均信息量。
- \* 一个随机变量的熵越大，它的不确定性越大，那么，正确估计其值的可能性越小。越不确定的随机变量越需要大的信息量用以确定其值。

# 模型评价：熵

- \* 举例：
  - \* 假设a, b, c, d, e, f 6个字符在某一简单的语言中随机出现，每个字符出现的概率分别为1/8, 1/4, 1/8, 1/4, 1/8, 1/8, 那么，每个字符的熵为：
  - \* 
$$H(p) = -\sum_{x \in \{a,b,c,d,e,f\}} P(x) \log P(x)$$
$$= -\left[4 \times \frac{1}{8} \log \frac{1}{8} + 2 \times \frac{1}{4} \log \frac{1}{4}\right] = 2\frac{1}{2}$$
  - \* 这个结果表明，我们可以设计一种编码，传输一个字符平均只需要2.5个比特：
- |         |    |     |    |     |     |
|---------|----|-----|----|-----|-----|
| 字符：a    | b  | c   | d  | e   | f   |
| 编码 :100 | 00 | 101 | 01 | 110 | 111 |

# 模型评价:相对熵 ( KL距离 )

- \* 相对熵又称为Kullback-Leibler差异, 或者简称为KL距离, 是衡量相同事件空间里两个概率分布相对差距的测度。两个概率分布 $p(x)$ 和 $q(x)$ 的相对熵定义为:

$$D(p \parallel q) = \sum_{x \in X} \log \frac{p(x)}{q(x)}$$

其中约定 $0 \log \left( \frac{0}{q} \right) = 0$ ,  $p \log \left( \frac{p}{0} \right) = \infty$

- \* 表示成期望值为:

$$D(p \parallel q) = E_p \left( \log \frac{p(X)}{q(X)} \right)$$

- \* 两个随机变量分布完全相同时,  $p=q$ , 其相对熵为0。当两个随机分布的差别增加时, 其相对熵期望也增大。

# 模型评价：交叉熵

- \* 交叉熵的概念是用来衡量估计模型与真实概率分布之间的差异情况的。
- \* 如果一个随机变量 $X \sim p(x)$ ， $q(x)$ 为用于近似 $p(x)$ 的概率分布，那么，随机变量 $X$ 和模型 $q$ 之间的交叉熵定义为：

$$\begin{aligned} H(X, q) &= H(x) + D(p \parallel q) \\ &= -\sum_x p(x) \log q(x) \\ &= E_p(\log \frac{1}{q(x)}) \end{aligned}$$

# 模型评价：交叉熵

- \* 可以定义语言  $L = (X) \sim p(x)$  与其模型  $q$  的交叉熵为：

$$H(L, q) = - \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{x_1^n} p(x_1^n) \log q(x_1^n)$$

其中， $x_1^n = x_1, x_2, \dots, x_n$  为  $L$  的词序列（样本），这里的词指样本中出现的任意符号单位，包括词汇、数字、标点等。 $p(x_1^n)$  为  $x_1^n$  的概率（理论值）， $q(x_1^n)$  为模型  $q$  对于  $x_1^n$  的概率估计值。

- \* 至此，仍然无法计算这个语言的交叉熵，因为不知道真实概率  $p(x_1^n)$ ，不过可以假设这种语言是理想的， $n$  趋于无穷大时，其全部单词的概率和为1。

# 模型评价：交叉熵

- \* 根据信息论的定理：假定语言L是稳态(stationary)遍历的(ergodic)随机过程，L与其模型q的交叉熵计算公式就变为：

$$H(L, q) = - \lim_{n \rightarrow \infty} \frac{1}{n} \log q(x_1^n)$$

- \* 由此，可以根据模型q和一个含有大量数据L的样本计算交叉熵。在设计模型q时，目的是使交叉熵最小，从而使模型最接近真实概率分布 $p(x)$ 。一般，在n足够大时（记为N），我们近似采用如下计算方法：

$$H(L, q) \approx - \frac{1}{N} \log q(x_1^N)$$



# 模型评价：困惑度

- \* 在设计语言模型时，我们通常用困惑度 (perplexity) 来代替交叉熵衡量语言模型的好坏，给定语言L的样本  $l_1^n = l_1, l_2, \dots, l_n$ ，L的困惑度  $PP_q$  定义为：

$$PP_q = 2^{H(L,q)} \approx 2^{-\frac{1}{n} \log(l_1^n)} = [q(l_1^n)]^{-\frac{1}{n}}$$

- \* 语言模型设计的任务就是寻找困惑度最小的模型，使其最接近真实语言的情况。
- \* 在自然语言处理中，我们所说的语言模型的困惑度通常是指语言模型对于测试数据的困惑度。