隐马尔科夫模型和词性标注

刘挺
哈工大信息检索研究室
2004年春

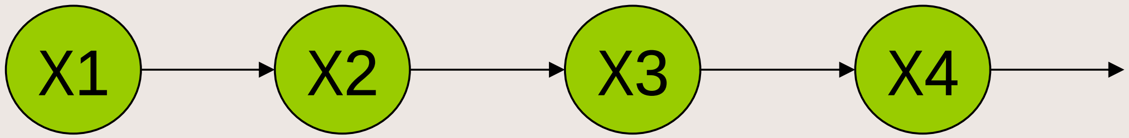
大纲

- 隐马尔科夫模型
  - 隐马尔科夫模型概述
  - 任务1：计算观察序列的概率
  - 任务2：计算能够解释观察序列的最大可能的状态序列
  - 任务3：根据观察序列寻找最佳参数模型
- 词性标注

隐马尔科夫模型概述

马尔科夫链

- 状态序列: $X_1, X_2, X_3, \ldots$
  - 常常是“时序”的
- 从X_{t-1}到X_t的转换只依赖于X_{t-1}

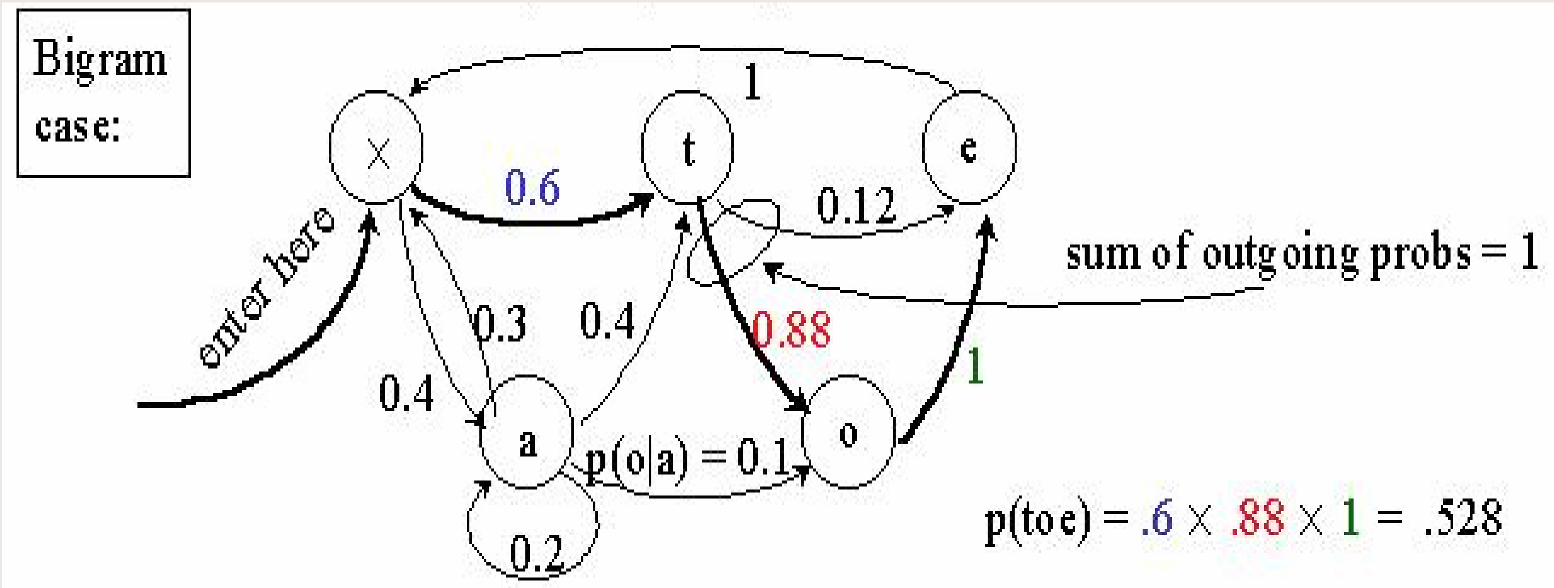


转移概率
Transition Probabilities

- 假设一个状态X_t有N个可能的值
  - $X_t=s_1$, $X_t=s_2$,….., $X_t=s_N$.
- 转移概率的数量为：N^2
  - $P(X_t=s_i|X_{t-1}=s_j)$, $1\leqslant i, j\leqslant N$
- 转移概率可以表示为N×N的矩阵或者有向图

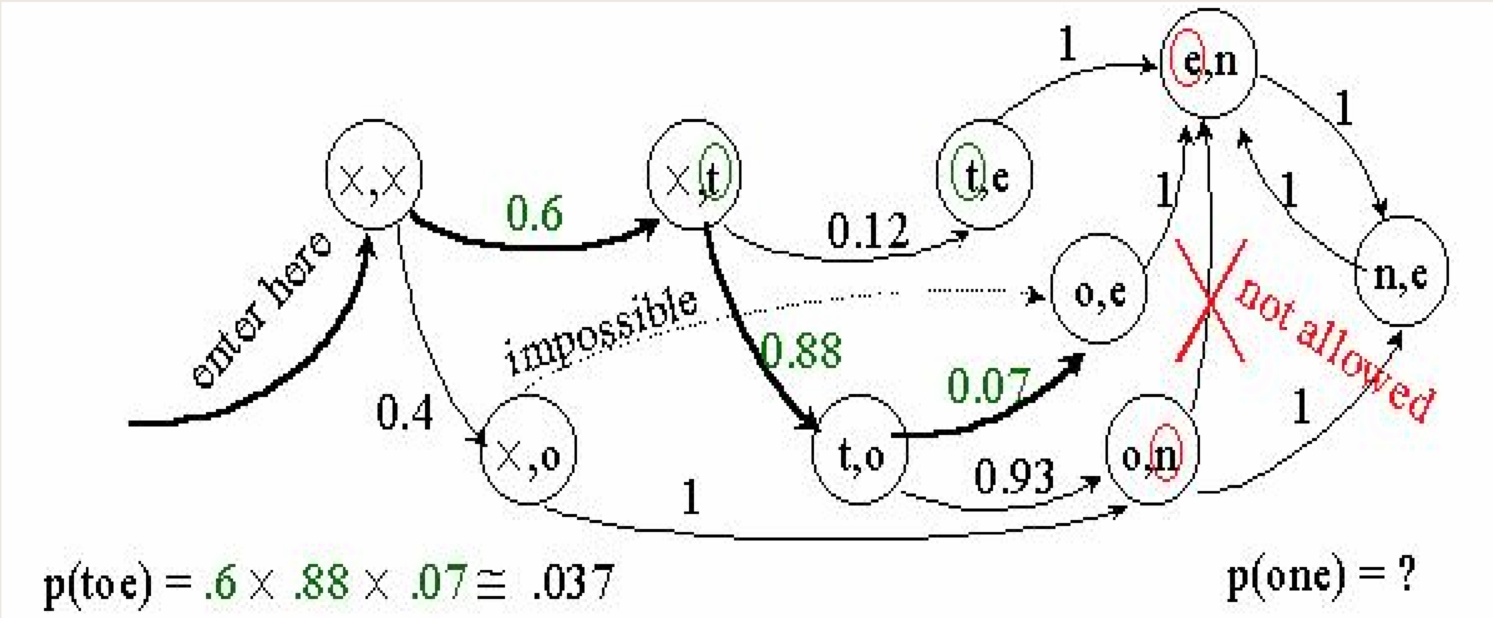
MM

- Bigram MM(一阶MM)



MM

- Trigram MM(二阶MM)

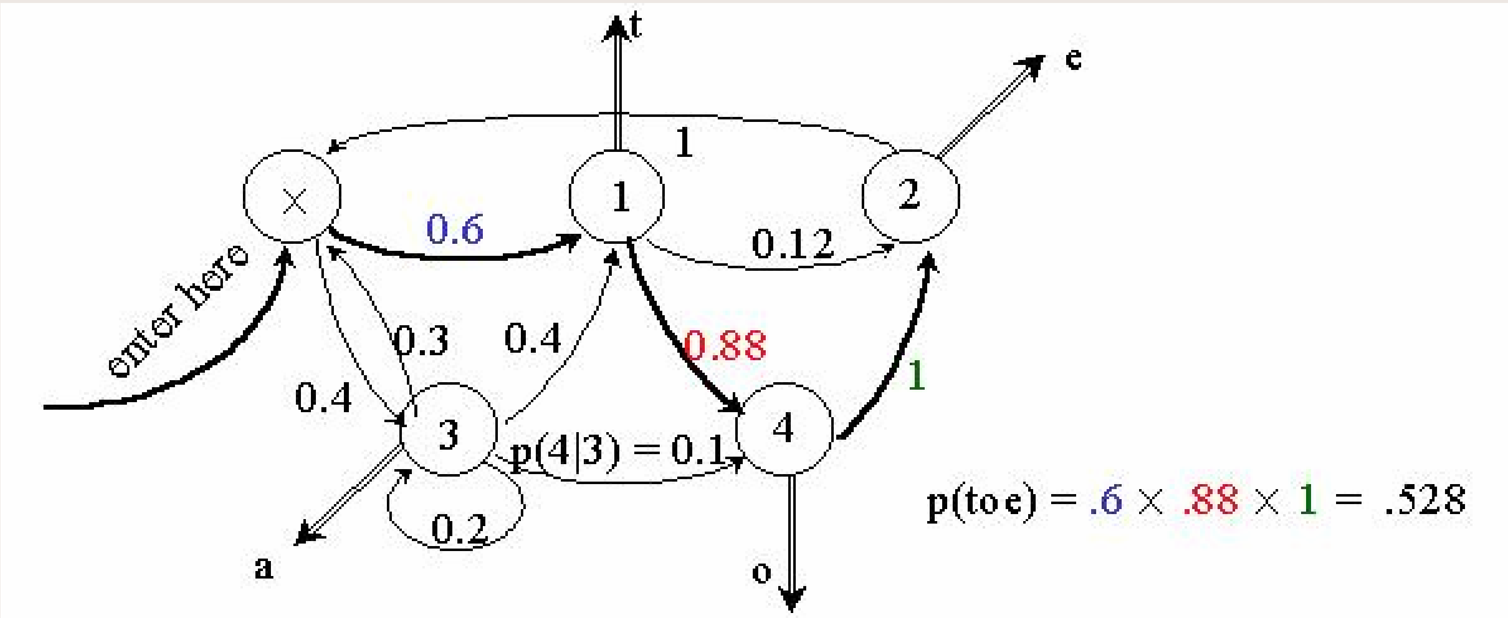


有限状态自动机

- 状态：输入输出字母表中的符号
- 弧：状态的转移
- 仍然是VMM (Visible MM)

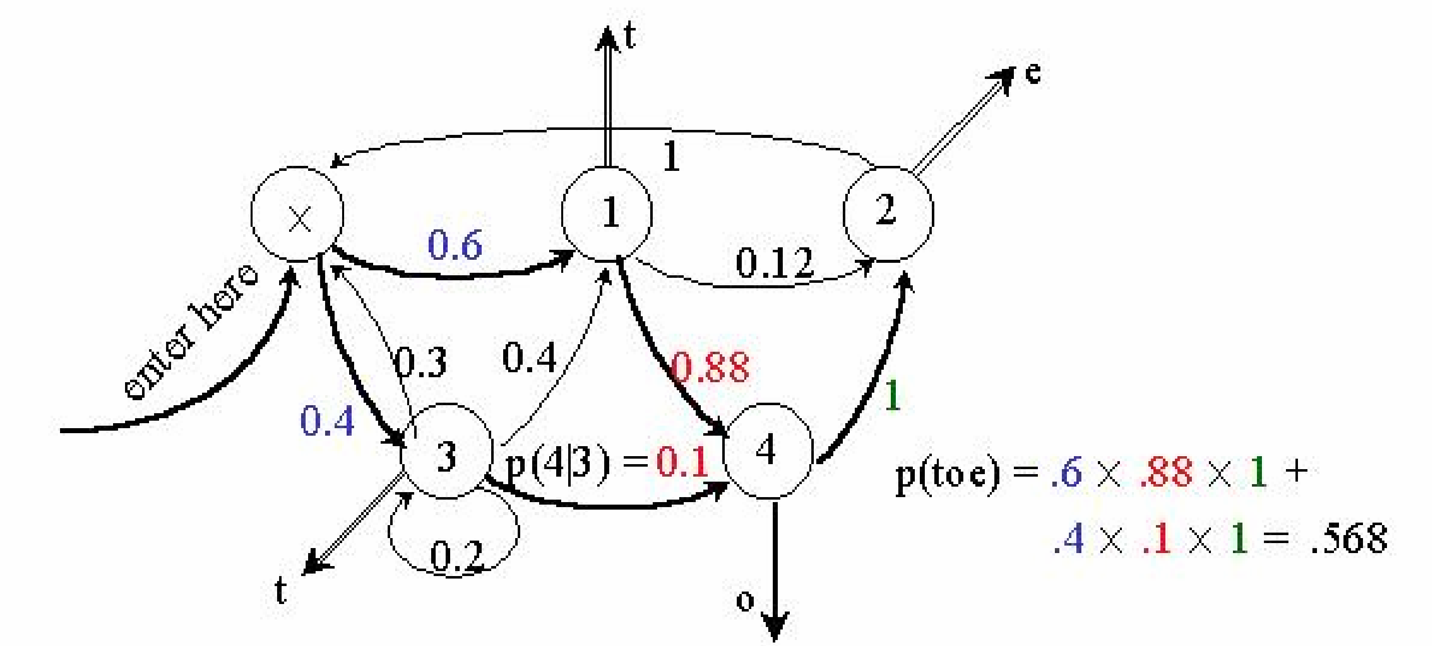
HMM

- HMM，从状态产生输出



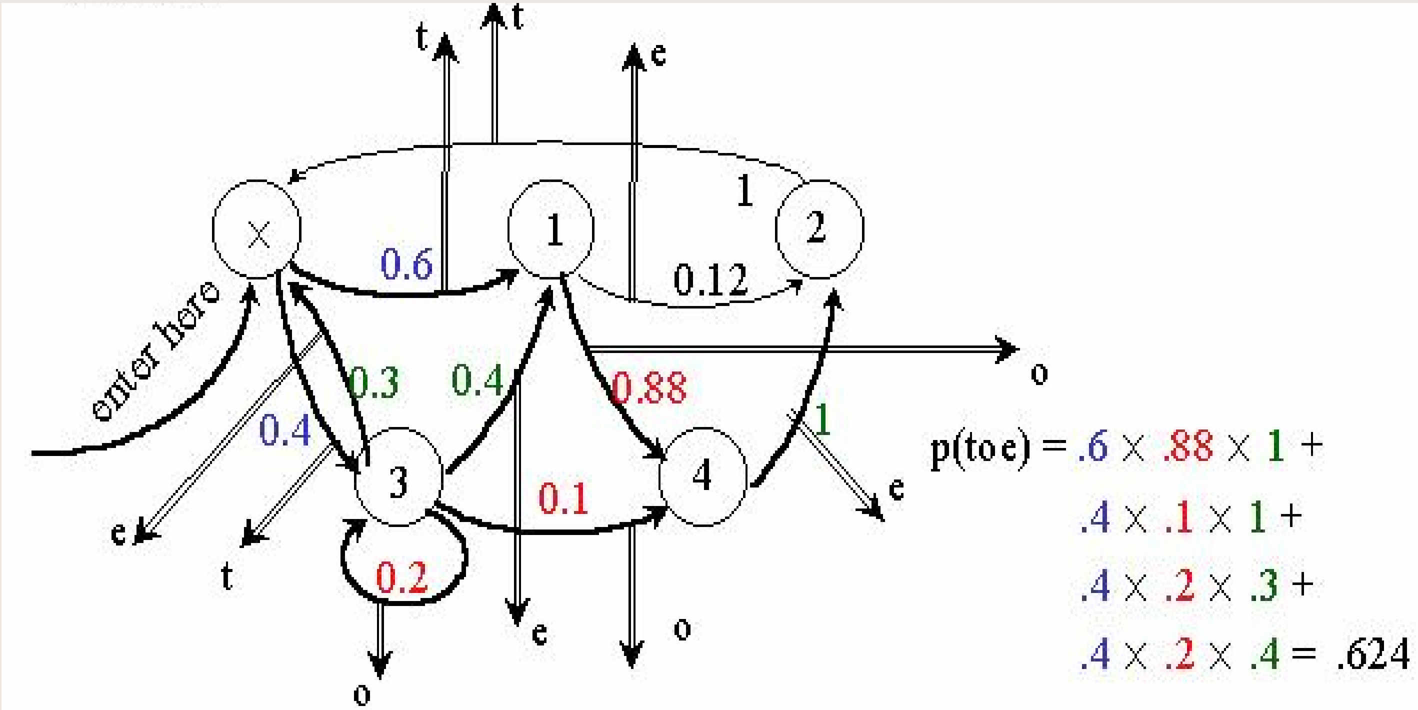
HMM

- HMM，不同状态可能产生相同输出



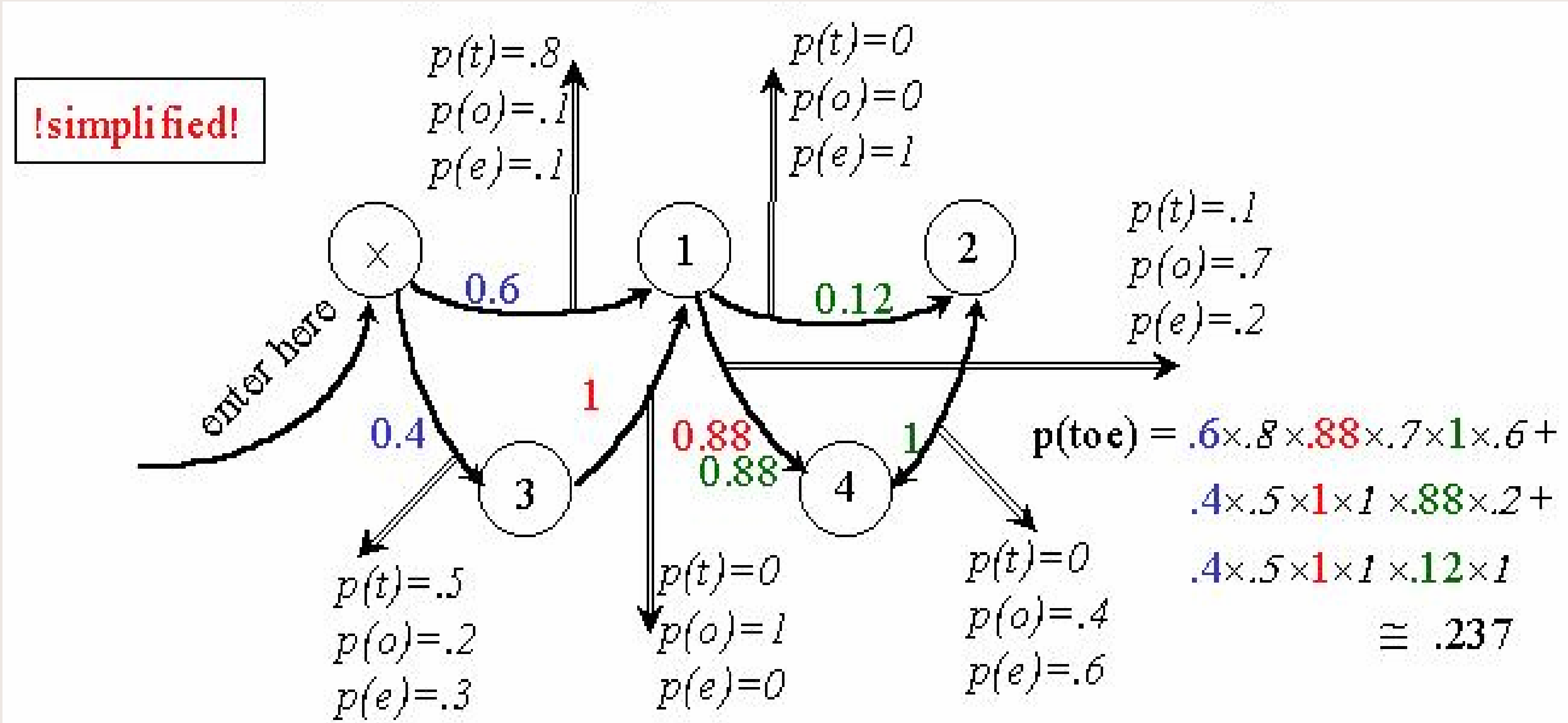
HMM

- HMM，从弧产生输出



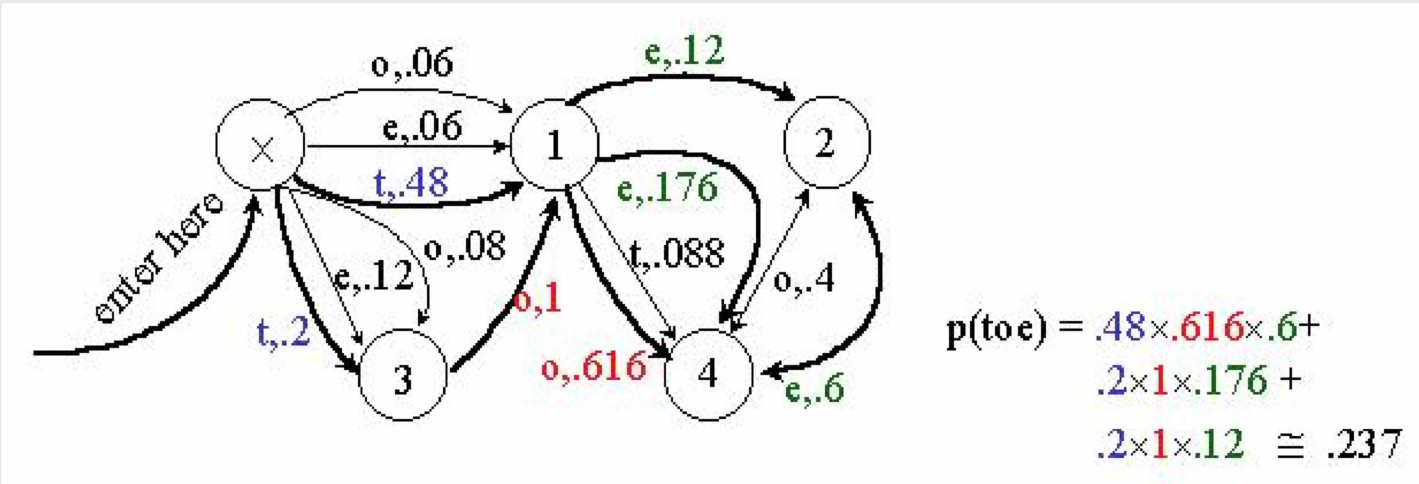
HMM

- HMM，输出带有概率



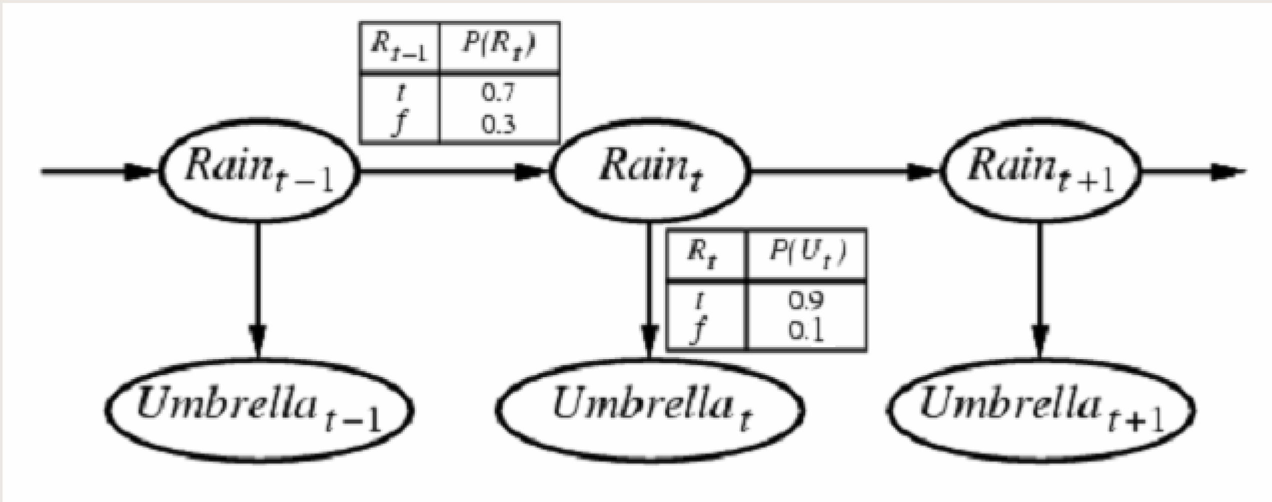
HMM

- HMM，两个状态间有多条弧，具有不同的概率



隐马尔可夫模型
Hidden Markov Model

- 估算隐藏于表面事件背后的事件的概率
  - 观察到一个人每天带雨伞的情况，反过来推测天气情况



Hidden Markov Model

- HMM是一个五元组(S, S_0,Y, Ps, P_Y).
  - S : {$s_1 \ldots s_T$ }是状态集，s_0是初始状态
  - Y : {$y_1 \ldots y_V$ }是输出字母表
  - $P_S(s_j|s_i)$:转移(transition)概率的分布，也表示为a_{ij}
  - $P_Y(y_k|s_i,s_j)$: 发射(emission)概率的分布，也表示为b_{ijk}

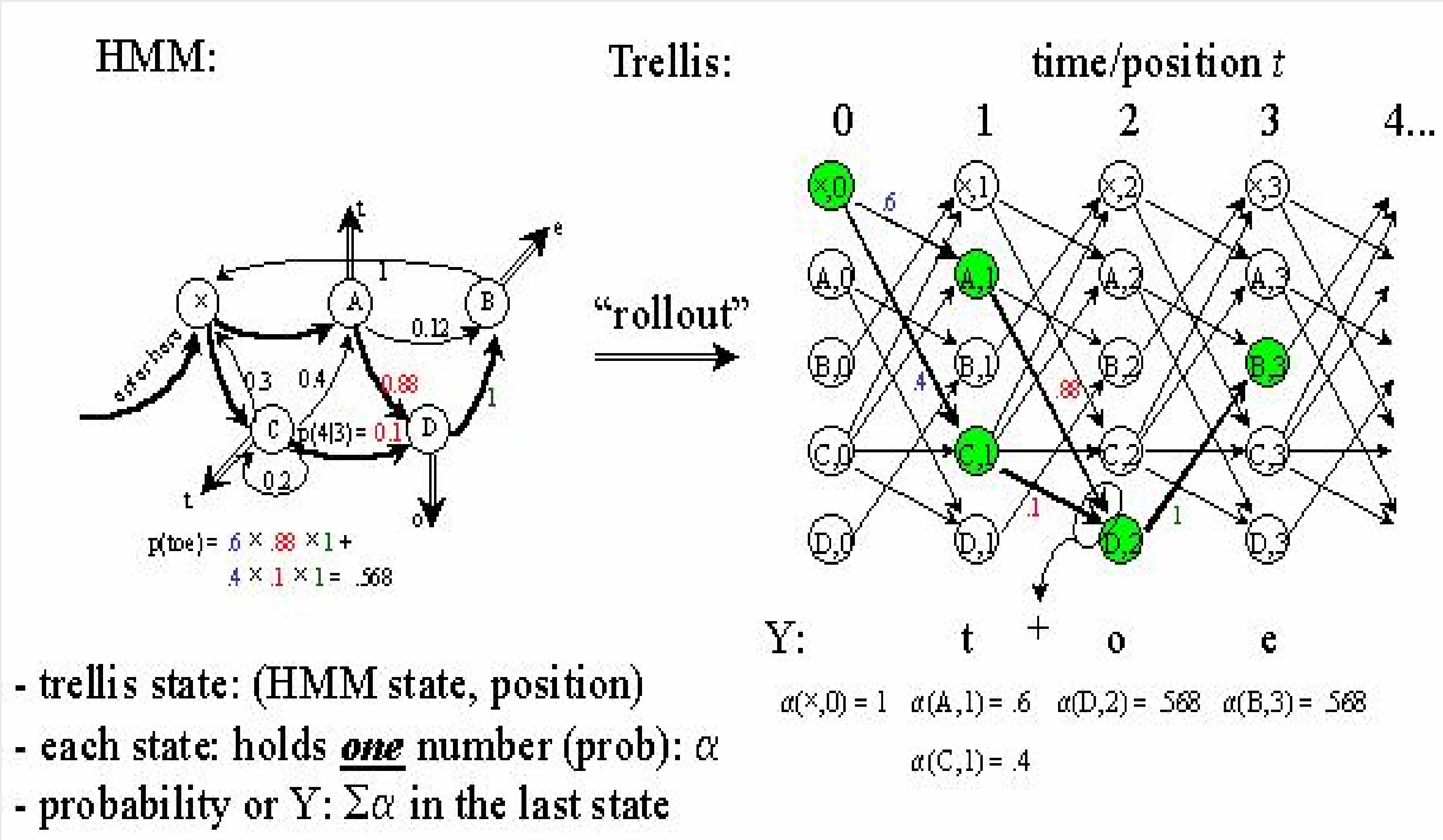
- 给定一个HMM和一个输出序列Y={$y_1,y_2,\ldots,y_k$)
  - 任务1：计算观察序列的概率
  - 任务2：计算能够解释观察序列的最大可能的状态序列
  - 任务3：根据观察序列寻找最佳参数模型

任务1：计算观察序列的概率

计算观察序列的概率

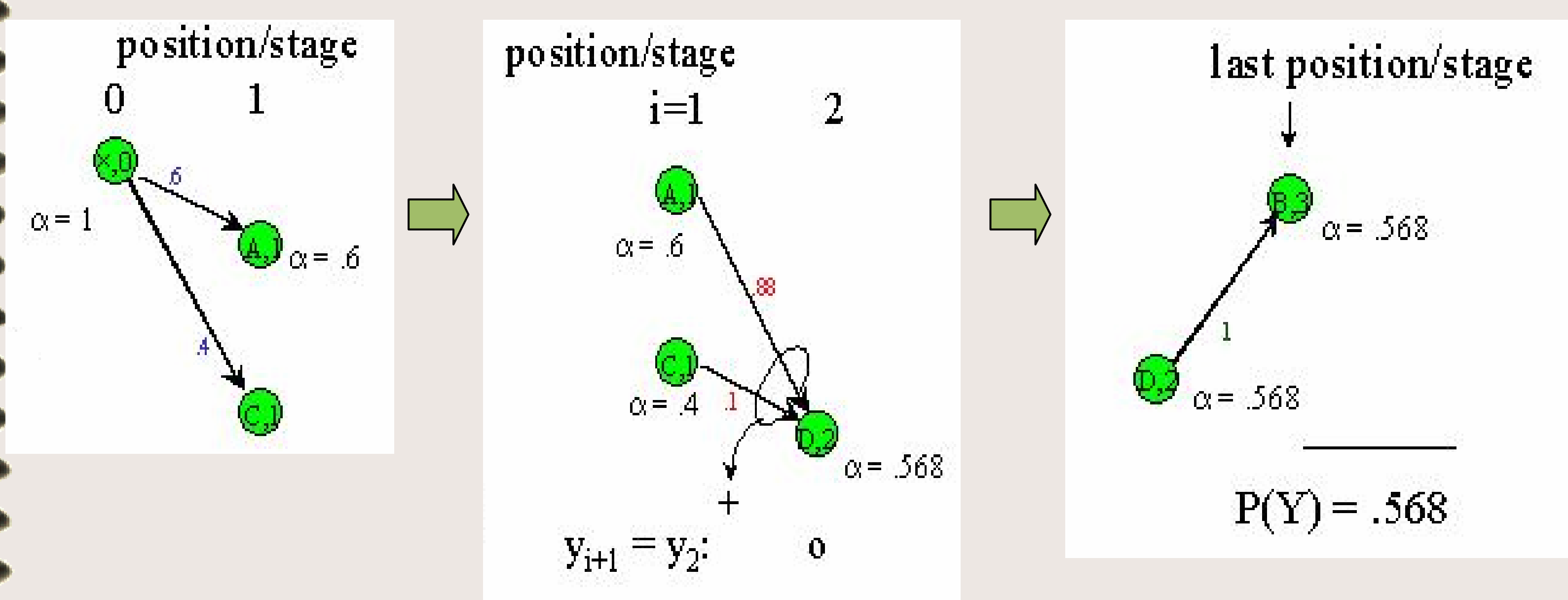
- 前提：HMM模型的参数已经训练完毕
- 想知道：根据该模型输出某一个观察序列的概率是多少
- 应用：基于类的语言模型，将词进行归类，变计算词与词之间的转移概率为类与类之间的转移概率，由于类的数量比词少得多，因此一定程度避免了数据稀疏问题

Trellis or Lattice(栅格)



发射概率为1的情况

- Y=“toe”
- P(Y)=0.6×0.88×1+0.4×0.1×1=0.568

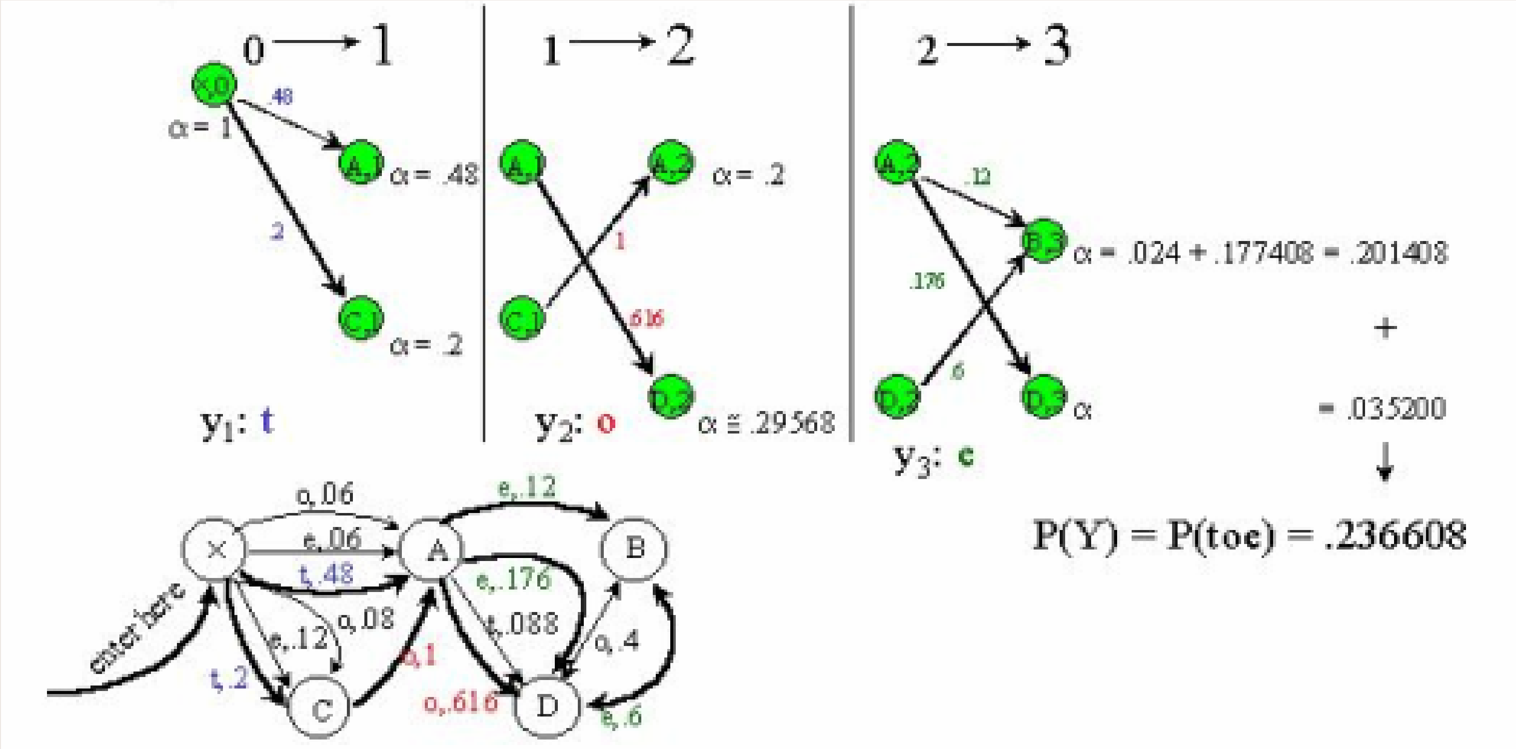


算法描述

- 从初始状态开始扩展
- 在时间点t扩展得到的状态必须能够产生于观察序列在t时刻相同的输出
  - 比如在t=1时，观察序列输出‘t’，因此只有状态A和C得到了扩展
- 在t+1时刻，只能对在t时刻保留下来的状态节点进行扩展
  - 比如在t=2时，只能对t=1时刻的A和C两个状态进行扩展
- 每条路径上的概率做累乘，不同路径的概率做累加
- 直到观察序列全部考察完毕，算法结束

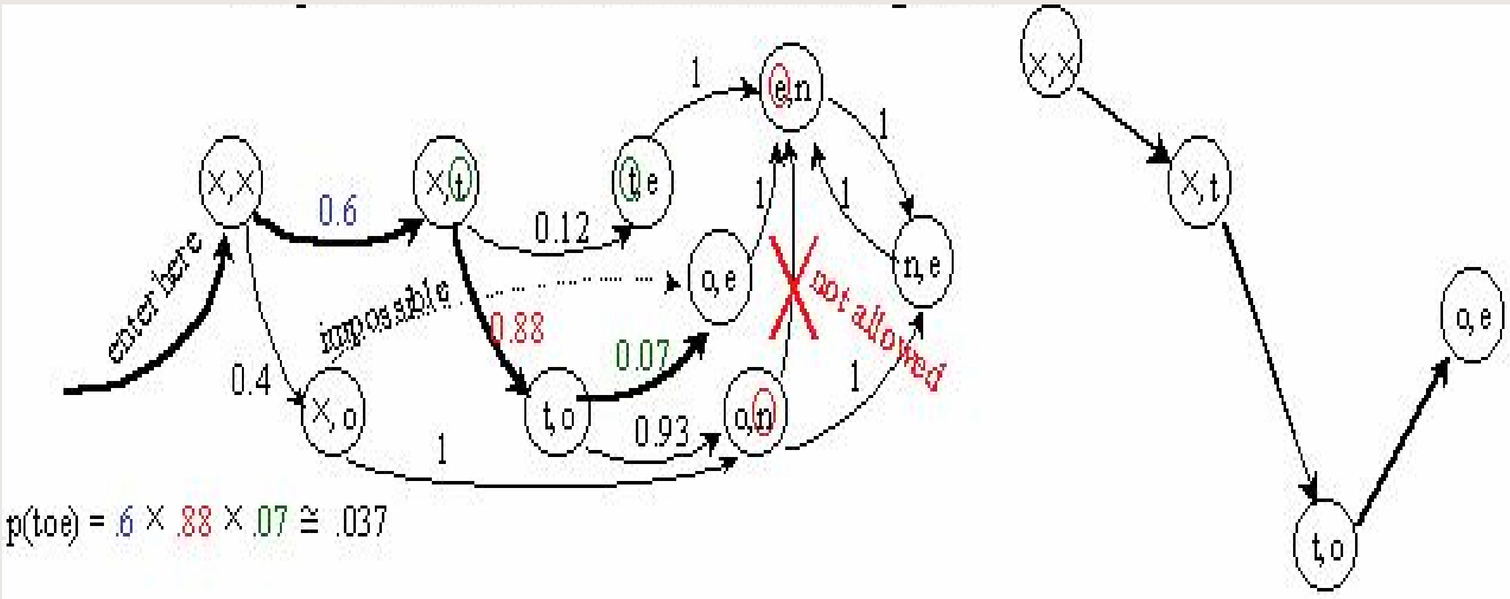
发射概率不为1的情况

- 0.236608就是在上述模型下“toe”出现的概率



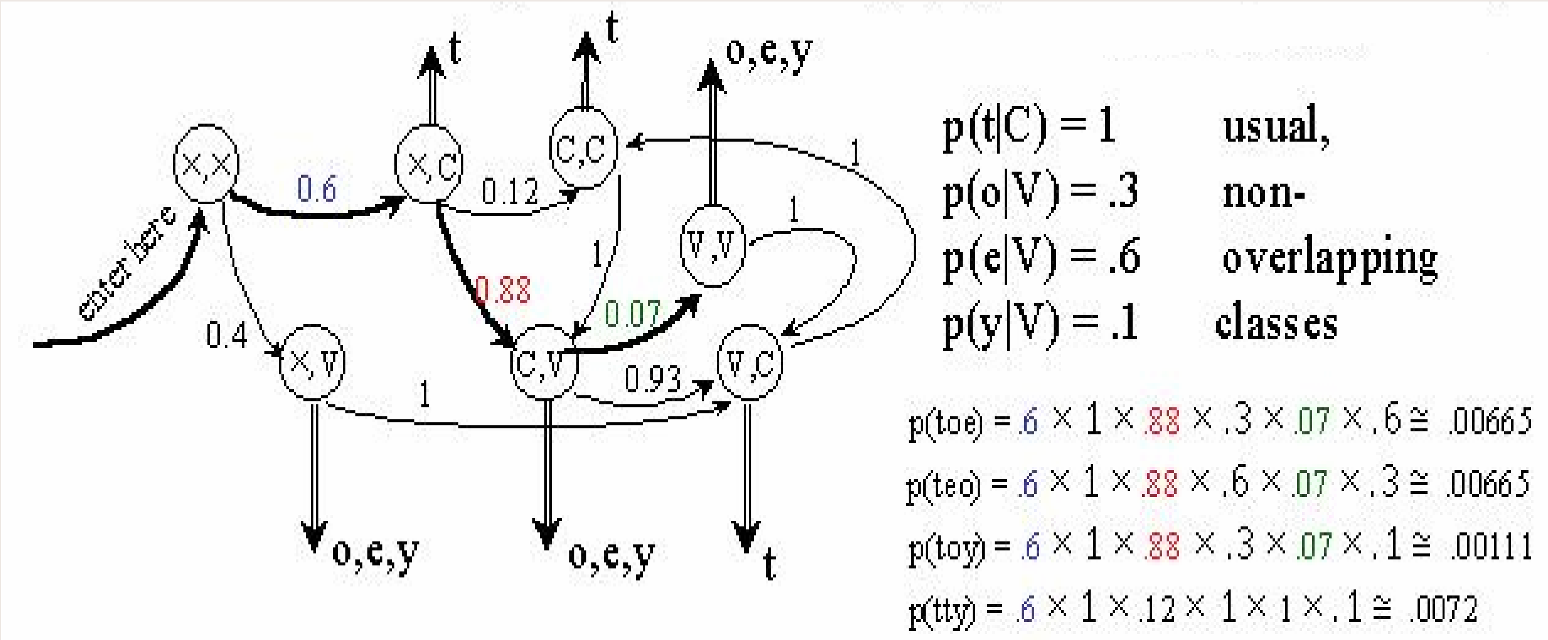
Trigram的情况

- 以Bigram为状态



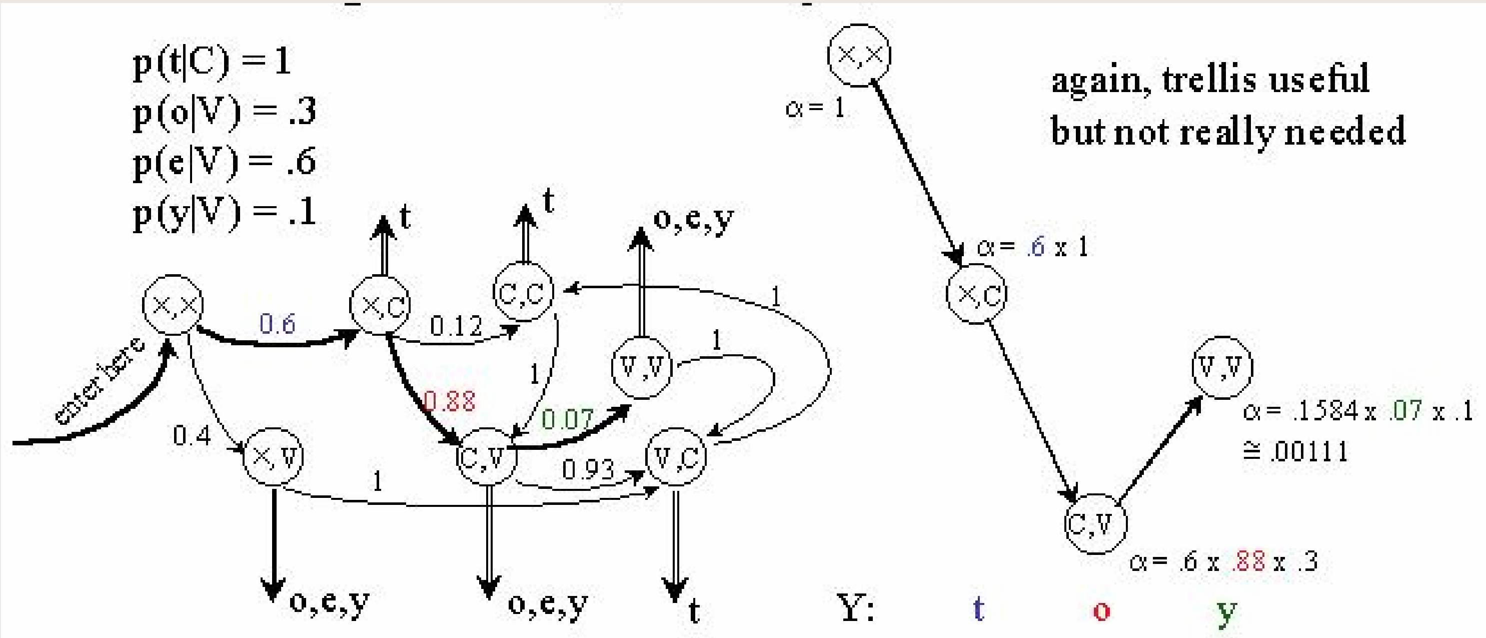
基于类的Trigram模型

- N-gram class LM
  - $p(w_i|w_{i-2},w_{i-1}) \rightarrow p(w_i|c_i)p(c_i|c_{i-2},c_{i-1})$
  - C:Consonant(辅音)，V:Vowel(元音)



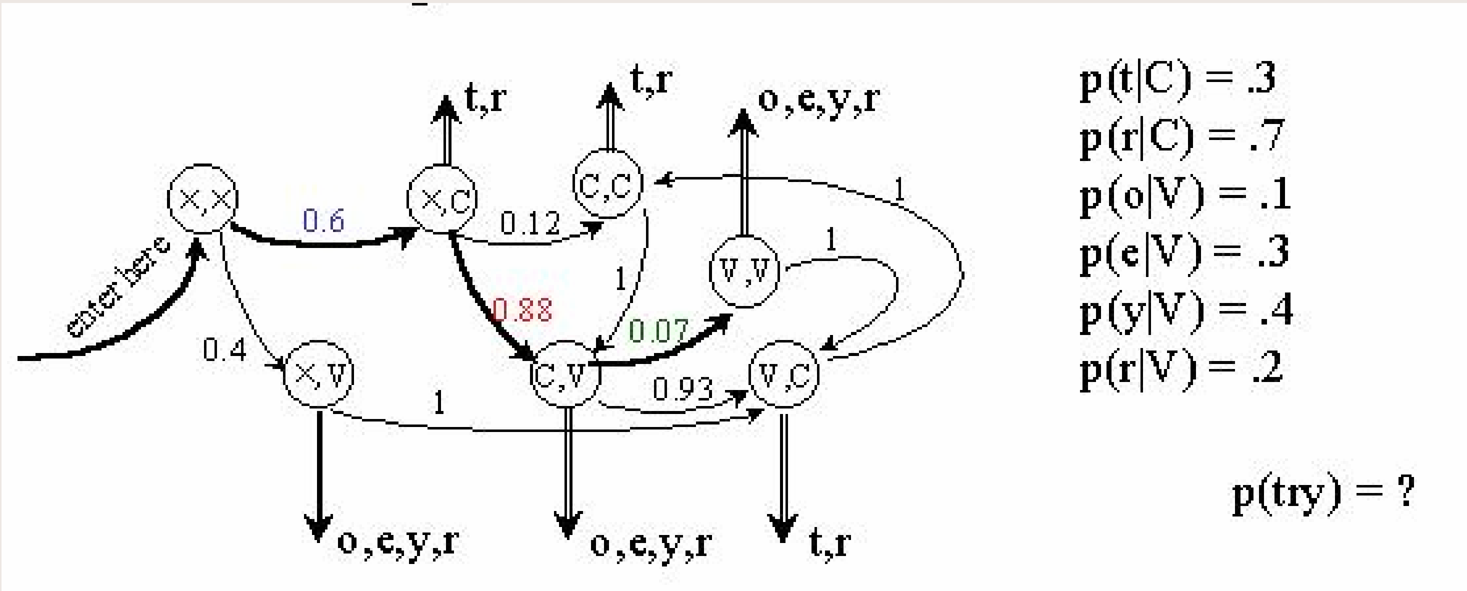
Class Trigram的Trellis

- 输出Y=“toy”

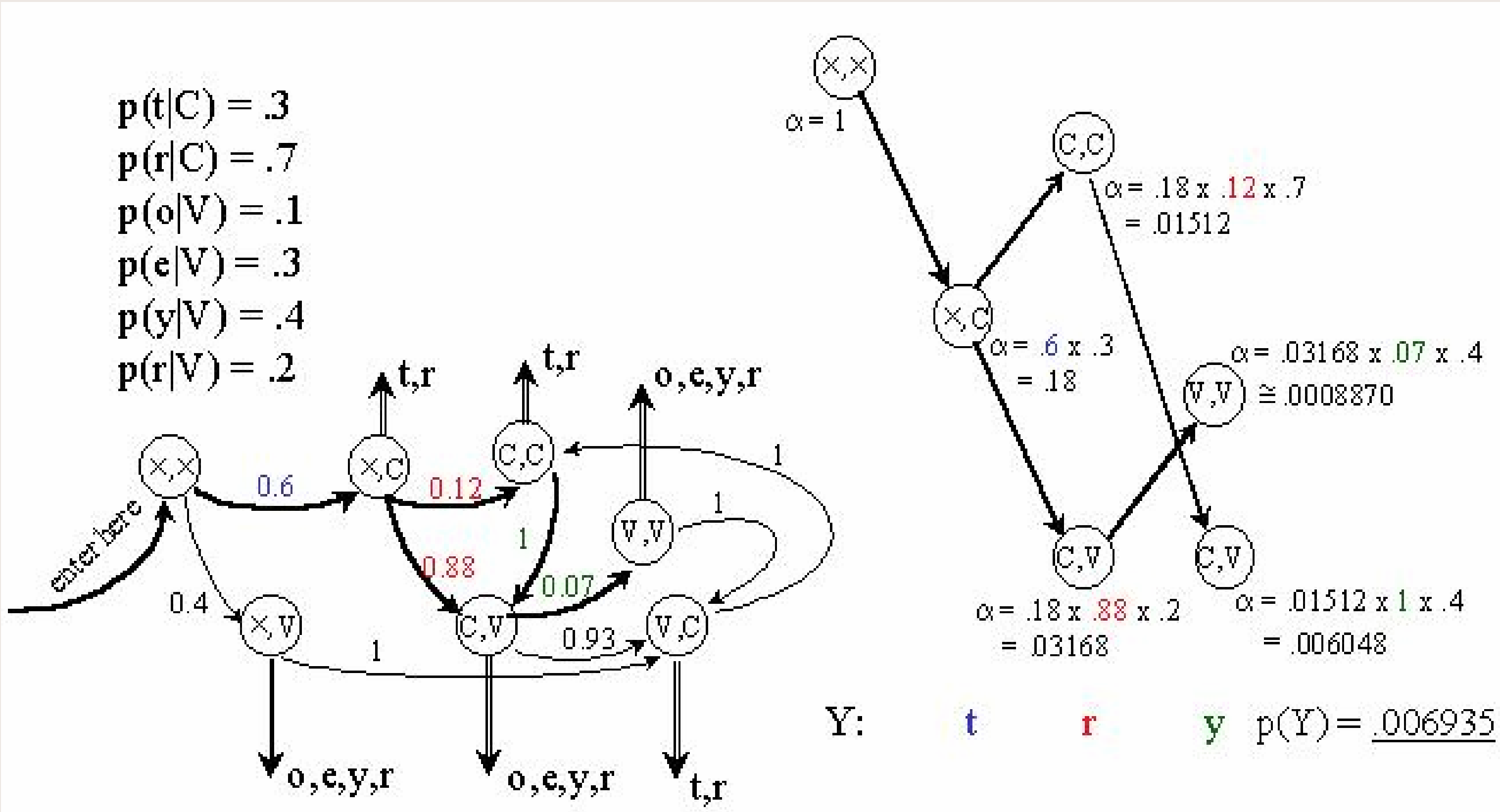


重叠(overlapping)的Class Trigram

- “r”有时是元音，有时是辅音，因此p(r|C)和p(r|V)都不为零



重叠的类Trigram的Trellis



讨论

- 我们既可以从左向右计算，也可以从右向左计算，甚至可以从中间向两头计算
- Trellis的计算对于Forward-Backward（也称为Baum-Welch)参数估计很有用

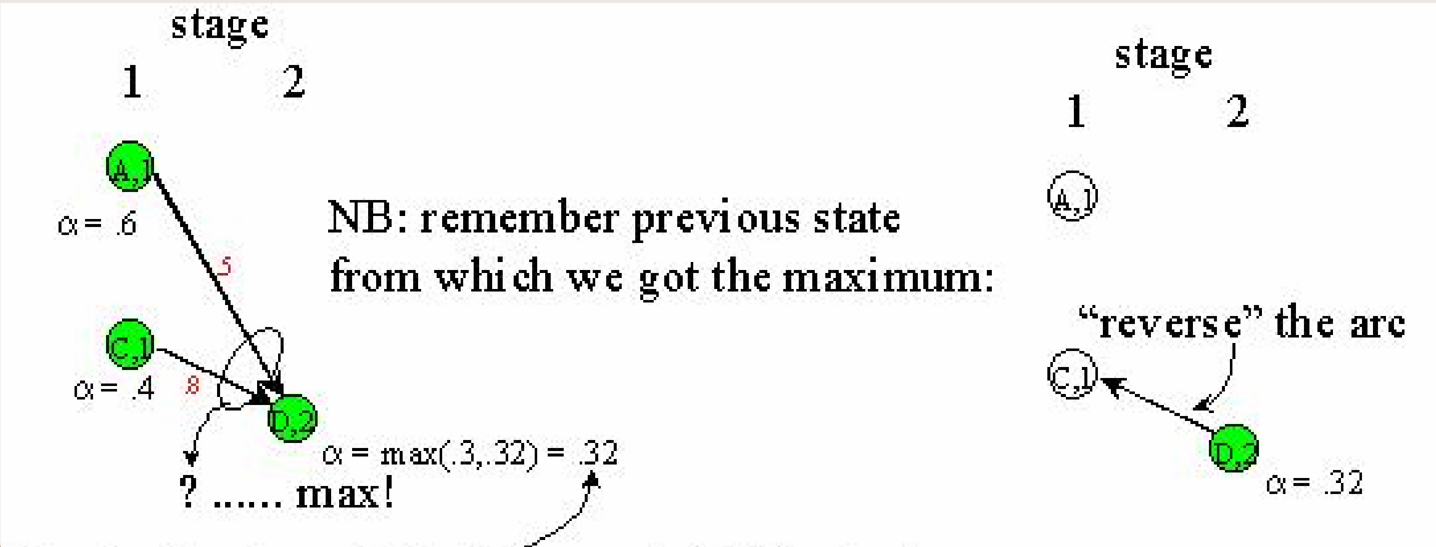
任务2：计算能够解释观察序列的最大可能的状态序列

Viterbi算法

- 用于搜索能够生成观察序列的最大概率的状态序列
- $S_{best}=\mathrm{argmax}_S P(S|Y)$
  $=\mathrm{argmax}_S P(S,Y)/P(Y)$
  $=\mathrm{argmax}_S \prod_{i=1\ldots k} p(y_i|s_i,s_{i-1})p(s_i|s_{i-1})$
- Viterbi能够找到最佳解，其思想精髓在于将全局最佳解的计算过程分解为阶段最佳解的计算

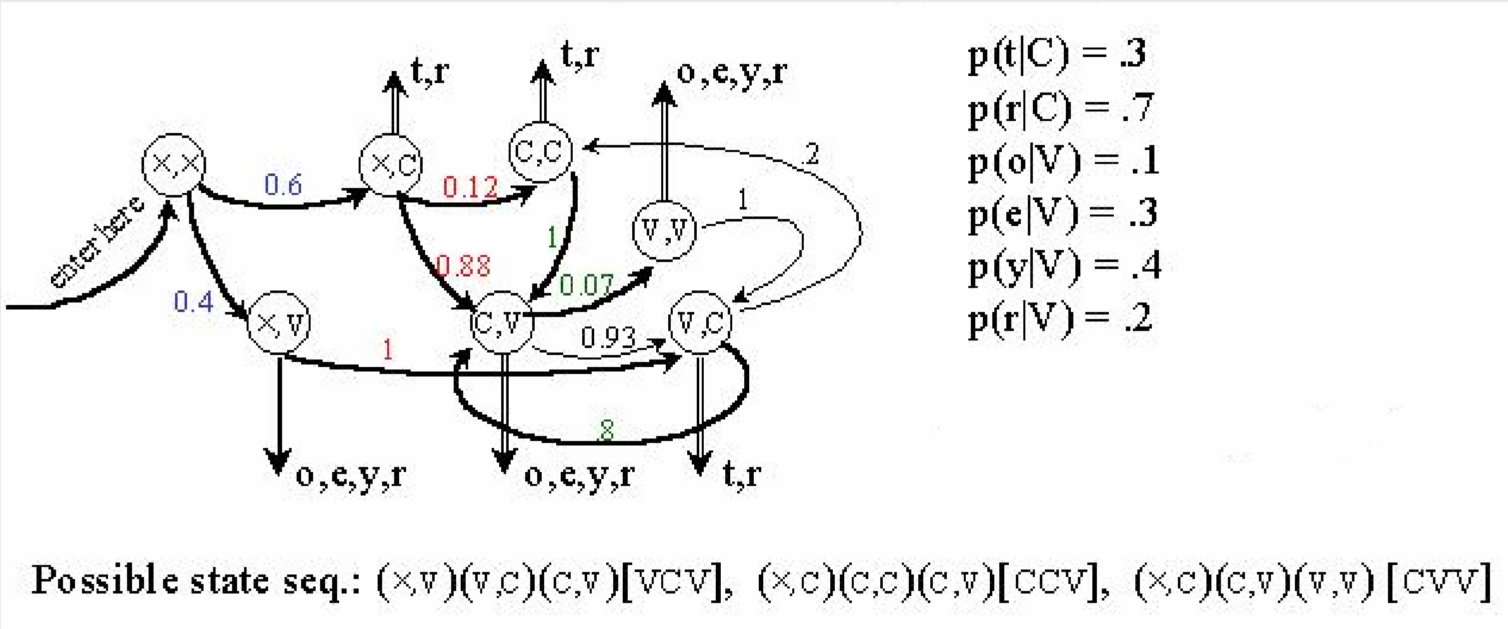
示意

- 从D2返回Stage 1的最佳状态为C1
  - 因为p(A1-D2)=0.6×0.5=0.3
  - 而p(C1-D2)=0.4×0.8=0.32
- 尽管搜索还没有完全结束，但是D2已经找到了最佳返回节点

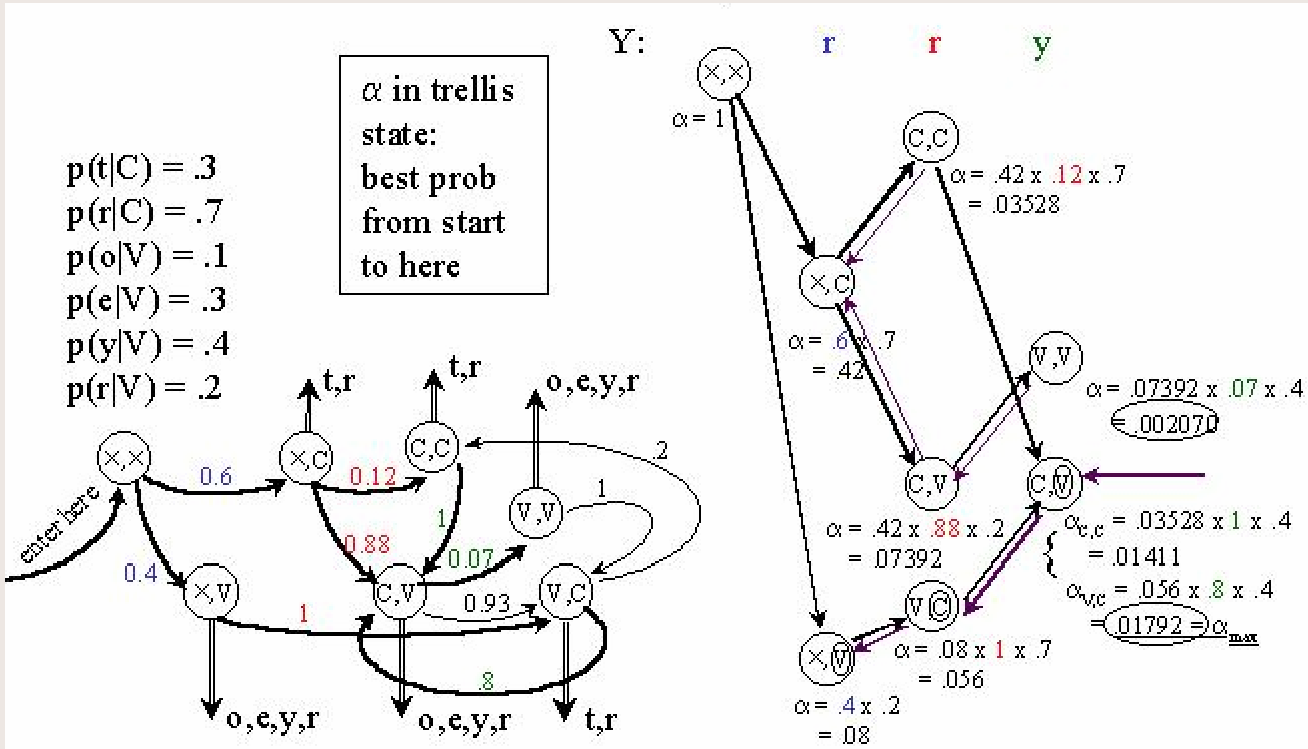


Viterbi示例

- $\text{argmax}_{XYZ}P(XYZ|rry)$



Viterbi计算



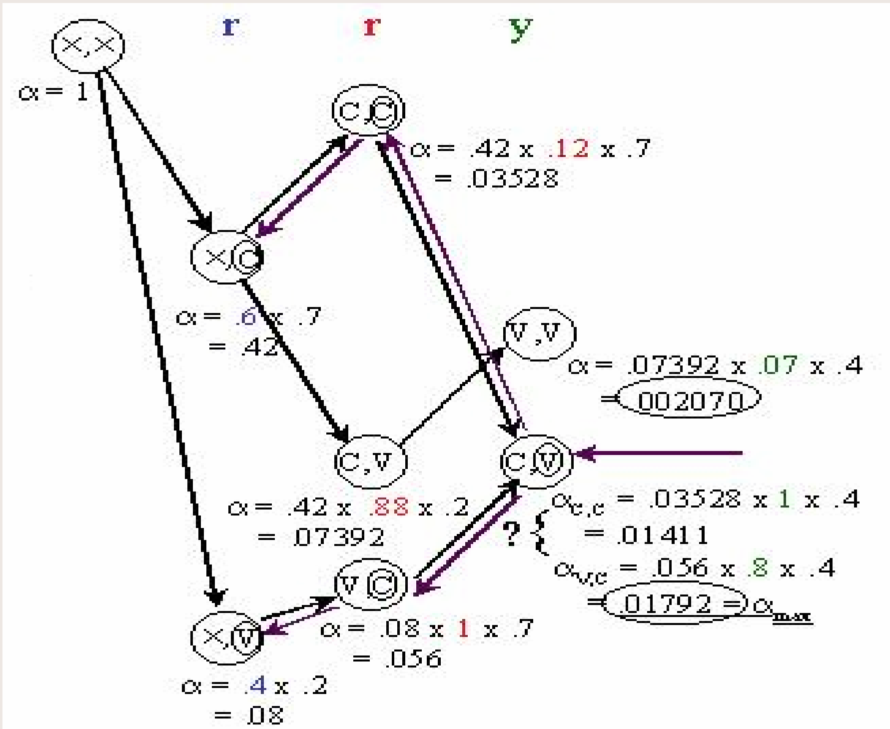
α in trellis state: best prob from start to here
p(t|C) = .3
p(r|C) = .7
p(o|V) = .1
p(e|V) = .3
p(y|V) = .4
p(r|V) = .2
enter here
t,r
o,e,y,r
Y: r r y
α = 1
α = .42 x .12 x .7 = .03528
α = .6 x .7 = .42
α = .07392 x .07 x .4 = .002070
α = .42 x .88 x .2 = .07392
α = .08 x 1 x .7 = .056
α = .4 x .2 = .08

Viterbi算法

- 三重循环
  - 第一重：遍历每一个观察值
  - 第二重：遍历当前观察值所对应的每一个状态
  - 第三重：遍历能够到达当前观察值当前状态的上一时刻的每一个状态
- 计算
  - 假设上一时刻为t，t时刻的的状态为i，t+1时刻的状态为j，t+1时刻的观察值为k，则计算：
    - $\delta_j(t+1)=\max_{1\le i\le N}\delta_i(t)a_{ij}b_{ijk}$
    - $\psi_j(t+1)=\arg\max_{1\le i\le N}\delta_i(t)a_{ij}b_{ijk}$
    - t+1时刻状态j的返回指针指向t时刻的状态$\psi_j(t+1)$
- 输出
  - 三重循环都结束后，在最后时刻找到δ值最大的状态，并从该状态开始，根据返回指针查找各时刻的处于最佳路径上的状态，并反序输出。

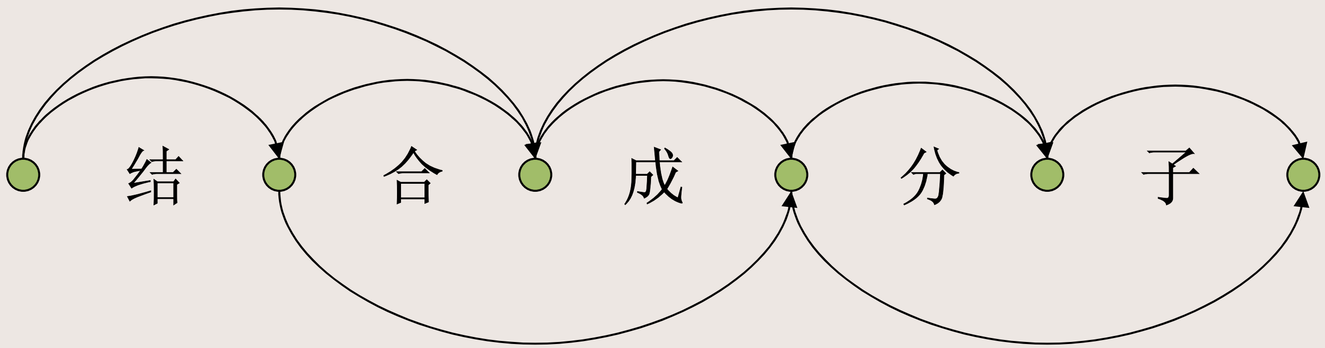
N-best计算

- 保留n个最佳结果，而不是1个
- 最优解：VCV；次优解：CCV



N-Best Paths

- 以分词为例（**MM**模型）
  - 例句：“结合成分子”
  - 每条弧上的值是该弧所对应的词的**Unigram**概率的负倒数，即**-logp(w)**



N-Best Paths

– **A sample**

The sentence “结合成分子 “.

● 结 ● 合 ● 成 ● 分 ● 子 ●

value	pre
0	0
0	0
0	0
0	0

value	Pre
∞	0
∞	0
∞	0
∞	0

value	pre
∞	0
∞	0
∞	0
∞	0

value	pre
∞	0
∞	0
∞	0
∞	0

value	pre
∞	0
∞	0
∞	0
∞	0

value	pre
∞	0
∞	0
∞	0
∞	0

N-Best Paths

– **A sample**

The sentence “结合成分子 “.

value	pre
0	0
0	0
0	0
0	0

value	Pre
10. 1	0
∞	0
∞	0
∞	0

value	pre
∞	0
∞	0
∞	0
∞	0

value	pre
∞	0
∞	0
∞	0
∞	0

value	pre
∞	0
∞	0
∞	0
∞	0

value	pre
∞	0
∞	0
∞	0
∞	0

N-Best Paths

– **A sample**

The sentence “结合成分子 “.

value	pre
0	0
0	0
0	0
0	0

value	Pre
10.1	0
∞	0
∞	0
∞	0

value	pre
7.76	0
∞	0
∞	0
∞	0

value	pre
∞	0
∞	0
∞	0
∞	0

value	pre
∞	0
∞	0
∞	0
∞	0

value	pre
∞	0
∞	0
∞	0
∞	0

N-Best Paths

– **A sample**

The sentence “结合成分子 “.

value	pre
0	0
0	0
0	0
0	0

value	Pre
10.1	0
∞	0
∞	0
∞	0

value	pre
7.76	0
20.0	1
∞	0
∞	0

value	pre
∞	0
∞	0
∞	0
∞	0

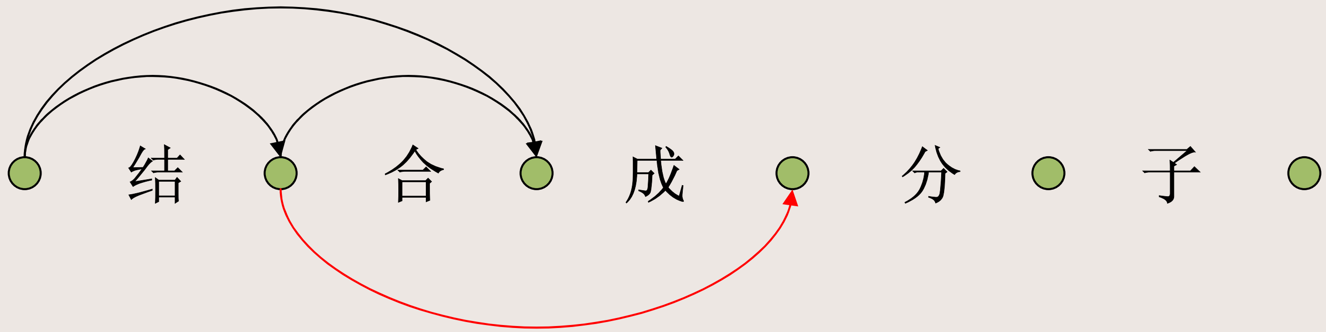
value	pre
∞	0
∞	0
∞	0
∞	0

value	pre
∞	0
∞	0
∞	0
∞	0

N-Best Paths

– **A sample**

The sentence “结合成分子 “.



value	pre
0	0
0	0
0	0
0	0

value	Pre
10.1	0
∞	0
∞	0
∞	0

value	pre
7.76	0
20.0	1
∞	0
∞	0

value	pre
21.5	1
∞	0
∞	0
∞	0

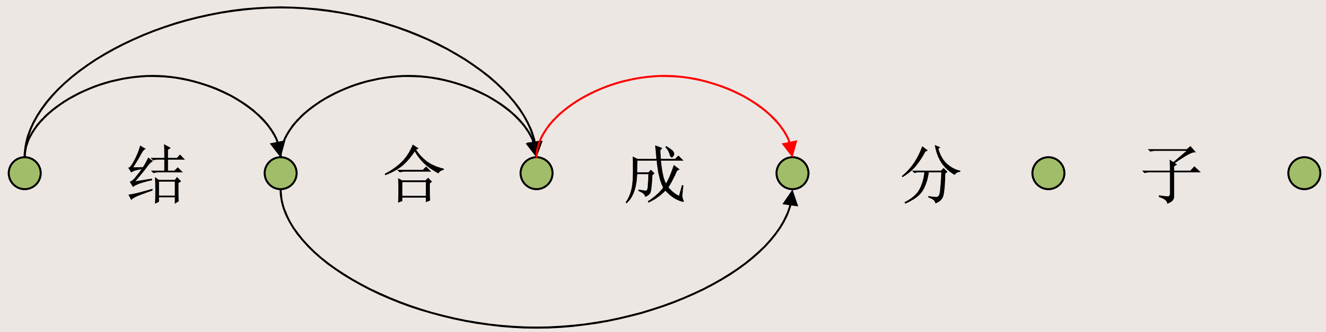
value	pre
∞	0
∞	0
∞	0
∞	0

value	pre
∞	0
∞	0
∞	0
∞	0

N-Best Paths

– **A sample**

The sentence “结合成分子 “.



value	pre
0	0
0	0
0	0
0	0

value	Pre
10.1	0
∞	0
∞	0
∞	0

value	pre
7.76	0
20.0	1
∞	0
∞	0

value	pre
14.4	2
21.5	1
27.6	2
∞	0

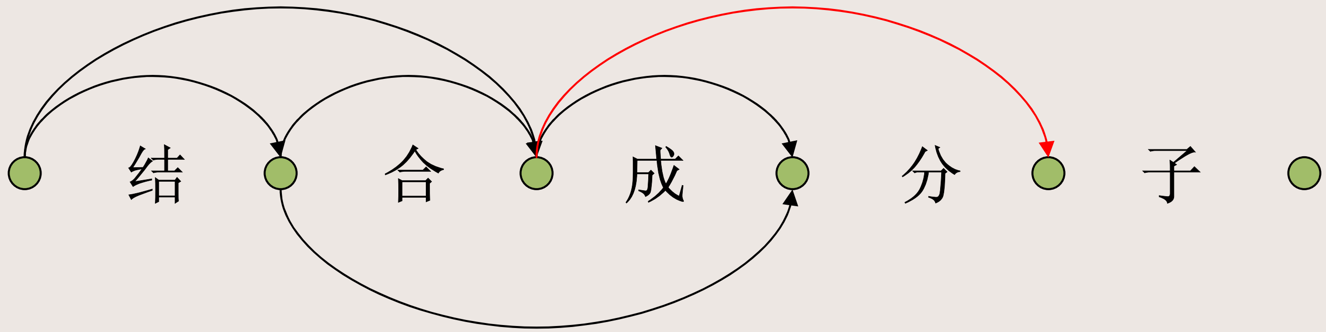
value	pre
∞	0
∞	0
∞	0
∞	0

value	pre
∞	0
∞	0
∞	0
∞	0

N-Best Paths

– **A sample**

The sentence “结合成分子 “.



value	pre
0	0
0	0
0	0
0	0

value	Pre
10.1	0
∞	0
∞	0
∞	0

value	pre
7.76	0
20.0	1
∞	0
∞	0

value	pre
14. 4	2
21. 5	1
27. 6	2
∞	0

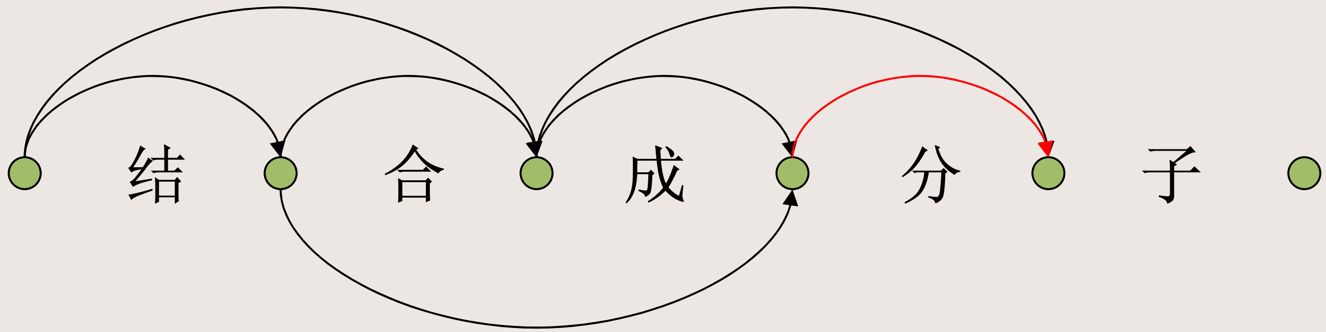
value	pre
18. 2	2
30. 5	2
∞	0
∞	0

value	pre
∞	0
∞	0
∞	0
∞	0

N-Best Paths

– **A sample**

The sentence “结合成分子 “.



value	pre
0	0
0	0
0	0
0	0

value	Pre
10.1	0
∞	0
∞	0
∞	0

value	pre
7.76	0
20.0	1
∞	0
∞	0

value	pre
14.4	2
21.5	1
27.6	2
∞	0

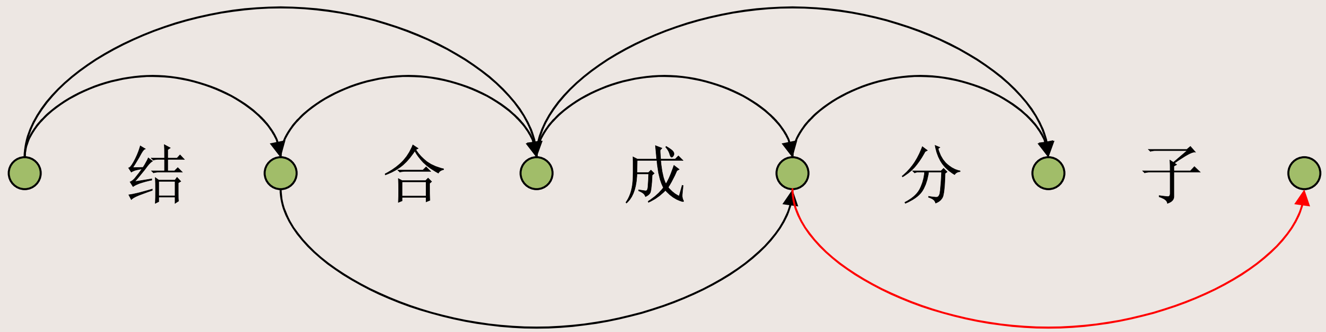
value	pre
18.2	2
23.4	3
30.0	3
30.5	2

value	pre
∞	0
∞	0
∞	0
∞	0

N-Best Paths

– **A sample**

The sentence “结合成分子 “.



value	pre
0	0
0	0
0	0
0	0

value	Pre
10.1	0
∞	0
∞	0
∞	0

value	pre
7.76	0
20.0	1
∞	0
∞	0

value	pre
14.4	2
21.5	1
27.6	2
∞	0

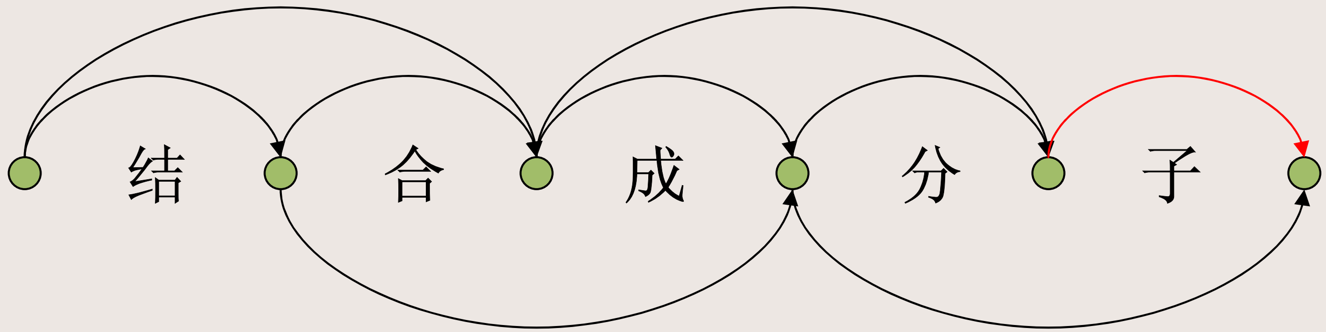
value	pre
18.2	2
23.4	3
30.0	3
30.5	2

value	pre
25.2	3
31.2	3
∞	0
∞	0

N-Best Paths

– **A sample**

The sentence “结合成分子 “.



value	pre
0	0
0	0
0	0
0	0

value	Pre
10.1	0
∞	0
∞	0
∞	0

value	pre
7.76	0
20.0	1
∞	0
∞	0

value	pre
14.4	2
21.5	1
27.6	2
∞	0

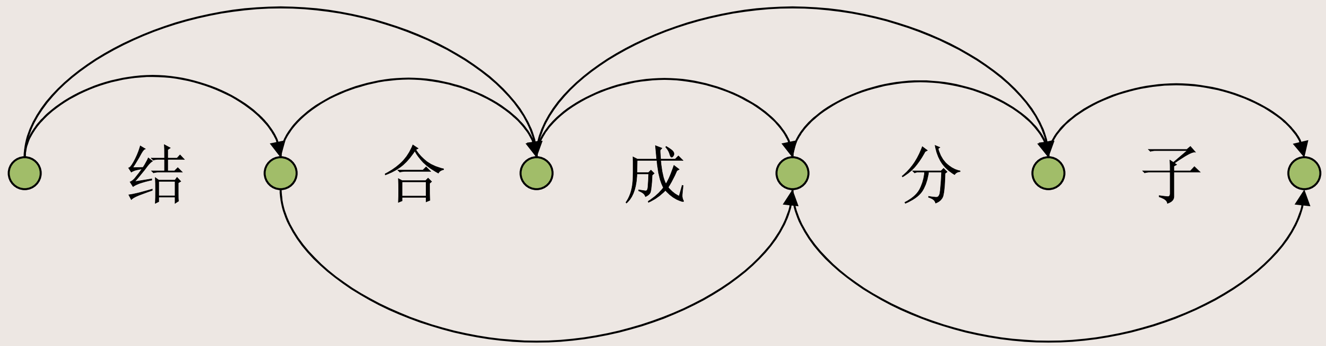
value	pre
18.2	2
23.4	3
30.0	3
30.5	2

value	pre
25.2	3
29.1	4
31.2	3
33.9	4

N-Best Paths

– **A sample**

The sentence “结合成分子 “.



value	pre
0	0
0	0
0	0
0	0

value	Pre
10.1	0
∞	0
∞	0
∞	0

value	pre
7.76	0
20. 0	1
∞	0
∞	0

value	pre
14.4	2
21.5	1
27. 6	2
∞	0

value	pre
18.2	2
23.4	3
30.0	3
30. 5	2

value	pre
25.2	3
29.1	4
31.2	3
33.9	4

结果

- 四条最佳路径为：
  1. 结合/成/分子
  2. 结合/成分/子
  3. 结/合成/分子
  4. 结合/成/分/子
- 时间复杂度
  - 假设搜索图中共有k条边
  - 要求获得N条最佳路径
  - 则时间复杂度为$O(k*N^2)$

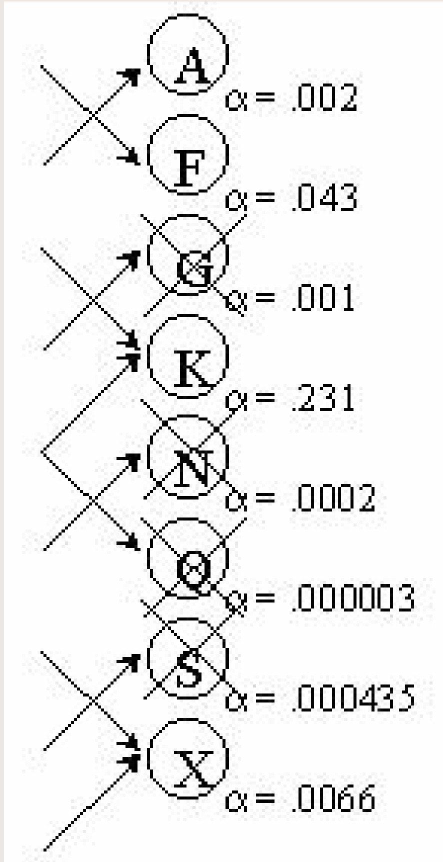
剪枝Pruning

在每一个时刻，如果Trellis上的状态过多，怎么办？

答案是剪枝：

1、按α的阈值剪枝， α太低的路径不再继续搜索

2、按状态的数量剪枝，超过多少个状态就不再扩展了



任务3：根据观察序列寻找最佳参数模型

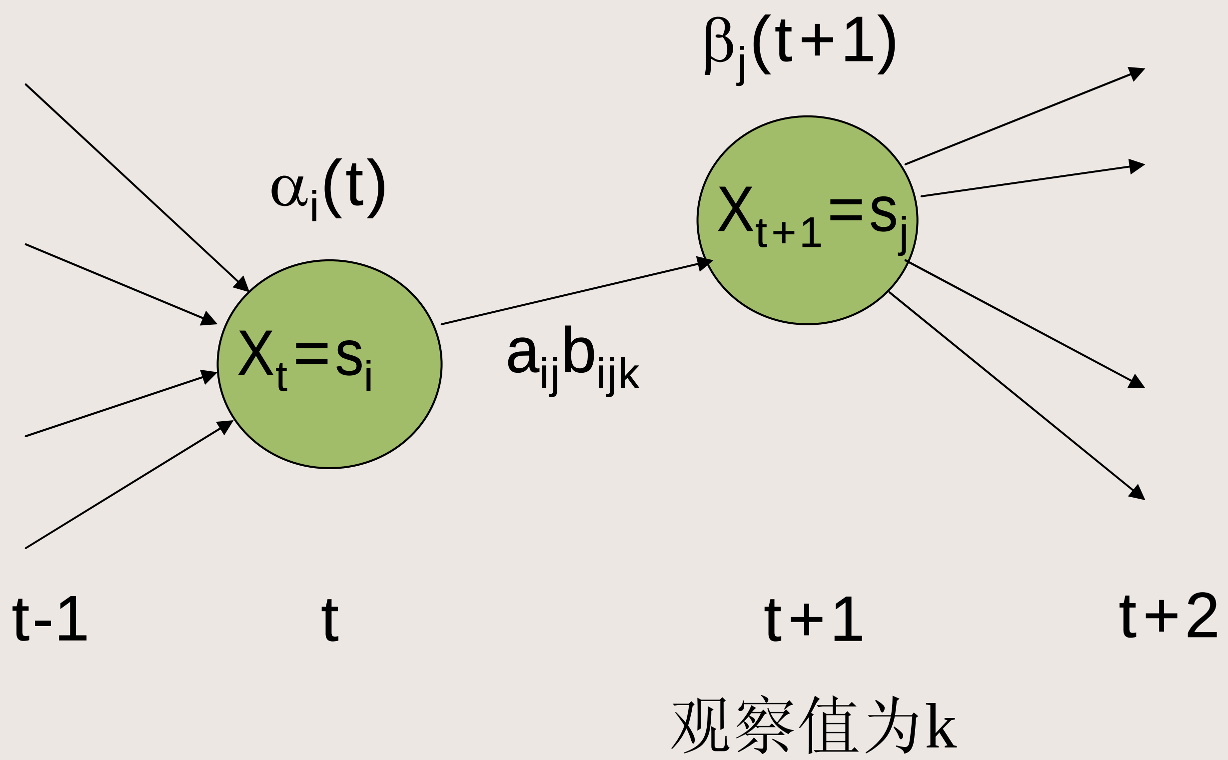
问题

- 给定一个观察值序列，但是没有标注每个观察值所对应的状态（无指导），在这种条件下如何估计隐马尔可夫模型中的参数，包括转移概率的分布和发射概率的分布
- 例如：给定一个语料库，语料库只是一个词的序列，没有词性标记，能否估计出词性标注的HMM模型？
- 是EM算法的特例，象一个魔法(MAGIC)！找到一个能够最佳地解释观察值序列的模型

Baum-Welch算法 也称为Forward-Backward算法

- 1. 初始化P_S，P_Y
  - 可能是随机给出的
- 2. 计算前向概率(Forward Probability)
  - $\alpha(s',i)=\sum_{s\to s'}\alpha(s,i-1)\times p(s'|s)\times p(y_i|s,s')$
  - 从左到右搜索过程中的累积值
- 3. 计算后向概率(Backward Probability)
  - $\beta(s',i)=\sum_{s'\leftarrow s}\beta(s,i+1)\times p(s|s')\times p(y_{i+1}|s',s)$
  - 从右到左搜索过程中的累积值

前向概率后向概率示意图



Baum-Welch算法（续）

- 4. 计数(pseudo count)
  - $c(y,s,s')=$
    - $\sum_{i=0\ldots k-1, y=yi+1} \alpha(s,i)p(s'|s)p(y_{i+1}|s,s')\beta(s',i+1)$
  - $c(s,s')=\sum_{y\in Y} c(y,s,s')$
  - $c(s)=\sum_{s\in S} c(s,s')$
- 5. 重新估算
  - $p'(s'|s)=c(s,s')/c(s)$, $p'(y|s,s')=c(y,s,s')/c(s,s')$
- 6. 重复运行2-5，直至结果不再有较大变化

词性标注

词性(Part of Speech)

- 词的句法类别
  - 名词、动词、形容词、副词、介词、助动词
  - 分为开放词类(Open Class)和封闭词类(Closed Class)
  - 也成为：语法类、句法类、POS标记、词类等

POS举例

开放类

N	noun	*baby, toy*
V	verb	*see, kiss*
ADJ	adjective	*tall, grateful, alleged*
ADV	adverb	*quickly, frankly, ...*
P	preposition	*in, on, near*
DET	determiner	*the, a, that*
WhPron	wh-pronoun	who, what, which, …
COORD	coordinator	*and, or*

替代性测试

- 两个词属于同一个词类，当且仅当它们相互替换时不改变句子的语法特征
  - The ______ is angry.（名词）
  - The _____ dog is angry.（形容词）
  - Fifi _____ .（不及物动词）
  - Fifi _____ the book.（及物动词）

POS Tags

- 不存在标准的词性标注集
  - 有的是用比较粗糙的标记集，例如：N, V, A, Aux, ….
  - 有的使用更细致的分类：(例如： Penn Treebank)
    - PRP: personal pronouns (you, me, she, he, them, him, her, …)
    - PRP$: possessive pronouns (my, our, her, his, …)
    - NN: singular common nouns (sky, door, theorem, …)
    - NNS: plural common nouns (doors, theorems, women, …)
    - NNP: singular proper names (Fifi, IBM, Canada, …)
    - NNPS: plural proper names (Americas, Carolinas, …)

Penn Treebank词性集

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	+,%, &
CD	Cardinal number	*one, two, three*	TO	"to"	*to*
DT	Determiner	*a, the*	UH	Interjection	*ah, oops*
EX	Existential 'there'	*there*	VB	Verb, base form	*eat*
FW	Foreign word	*mea culpa*	VBD	Verb, past tense	*ate*
IN	Preposition/sub-conj	*of, in, by*	VBG	Verb, gerund	*eating*
JJ	Adjective	*yellow*	VBN	Verb, past participle	*eaten*
JJR	Adj., comparative	*bigger*	VBP	Verb, non-3sg pres	*eat*
JJS	Adj., superlative	*wildest*	VBZ	Verb, 3sg pres	*eats*
LS	List item marker	*1, 2, One*	WDT	Wh-determiner	*which, that*
MD	Modal	*can, should*	WP	Wh-pronoun	*what, who*
NN	Noun, sing. or mass	*llama*	WP$	Possessive wh-	*whose*
NNS	Noun, plural	*llamas*	WRB	Wh-adverb	*how, where*
NNP	Proper noun, singular	*IBM*	$	Dollar sign	$
NNPS	Proper noun, plural	*Carolinas*	#	Pound sign	#
PDT	Predeterminer	*all, both*	"	Left quote	(' or ")
POS	Possessive ending	*'s*	"	Right quote	(' or ")
PP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PP$	Possessive pronoun	*your, one's*	)	Right parenthesis	( ],), }, >)
RB	Adverb	*quickly, never*	,	Comma	,
RBR	Adverb, comparative	*faster*	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	*fastest*	:	Mid-sentence punc	(: ; ... – -)
RP	Particle	*up, off*			

PRP
PRP$

词性标注

- 词常常有多个词性，以*back*为例
  - The *back* door = JJ
  - On my *back* = NN
  - Win the voters *back* = RB
  - Promised to *back* the bill = VB
- 词性标注问题就是针对确定词在一个特定实例中的词性

POS歧义 (在Brown语料库中)

无歧义的词(1 tag): 35,340个

有歧义的词 (2-7 tags): 4,100个

2 tags	3,760
3 tags	264
4 tags	61
5 tags	12
6 tags	2
7 tags	1

(Derose, 1988)

词性标注的应用

- 文娱转换
  - 怎样朗读"lead"
    - 动词一般形式：[li:d]
    - 过去式：[led]
- 是句法分析的基础
- 辅助词义消歧
  - 等，动词→等待
  - 等，量词→等级

目前的性能

- 容易评价，只需计算标注正确的词性数量
  - 目前准确率大约在97%左右
  - Baseline也可以达到90%
  - Baseline算法:
    - 对每一个词用它的最高频的词性进行标注
    - 未登录词全部标为名词

词性标注

- $P(T|W)=P(W|T)P(T)/P(W)$
- $\text{argmax}_T p(T|W)=\text{argmax}_T p(W|T)p(T)$
- $P(W|T)=\prod_{i=1\ldots d} p(w_i|w_1,\ldots,w_{i-1},t_1,\ldots,t_d)$
  - $p(w_i|w_1,\ldots,w_{i-1},t_1,\ldots,t_d) \cong p(w_i|t_i)$
- $P(T)=\prod_{i=1\ldots d} p(t_i|t_1,\ldots,t_{i-1})$
  - $p(t_i|t_1,\ldots,t_{i-1})=p(t_i|t_{i-n+1},\ldots,t_{i-1})$

有指导的学习

- 训练时事先对语料库进行了人工的词性标注，因此在训练时看到了状态（词性），属于VMM，在测试时，只能看到观察值（词序列），因此属于HMM。
- 应用最大似然估计
  - $p(w_i|t_i)=c_{wt}(t_i,w_i)/c_t(t_i)$
  - $p(t_i|t_{i-n+1},\ldots,t_{i-1})$
  - $=c_{tn}(t_{i-n+1},\ldots,t_{i-1},t_i)/c_{t(n-1)}(t_{i-n+1},\ldots,t_{i-1})$
- 平滑
  - $p(w_i|t_i)$：加1平滑
  - $p(t_i|t_{i-n+1},\ldots,t_{i-1})$：线性差值

用带标记的语料进行训练

- **Pierre/NNP Vinken/NNP , , 61/CD years/NNS old/JJ ,/, will/MD join/VB the/DT board/NN as/IN a/DT nonexecutive/JJ director/NN Nov./NNP 29/CD ./.**
- **Mr./NNP Vinken/NNP is/VBZ chairman/NN of/IN Elsevier/NNP N.V./NNP ,/, the/DT Dutch/NNP publishing/VBG group/NN . .**
- **Rudolph/NNP Agnew/NNP ,/, 55/CD years/NNS old/JJ and/CC former/JJ chairman/NN of/IN Consolidated/NNP Gold/NNP Fields/NNP PLC/NNP ,/, was/VBD named/VBN a/DT nonexecutive/JJ director/NN of/IN this/DT British/JJ industrial/JJ conglomerate/NN ./.**

c(JJ)=7 c(JJ, NN)=4, P(NN|JJ)=4/7

无指导的学习

- 语料库只是词的序列，没有人工标注词性，是Plain Text。
- 完全无指导的学习是不可能的
  - 至少要知道：
    - 词性集
    - 每个词可能的词性（据词典）
- 使用Baum-Welch算法

无指导学习的秘诀

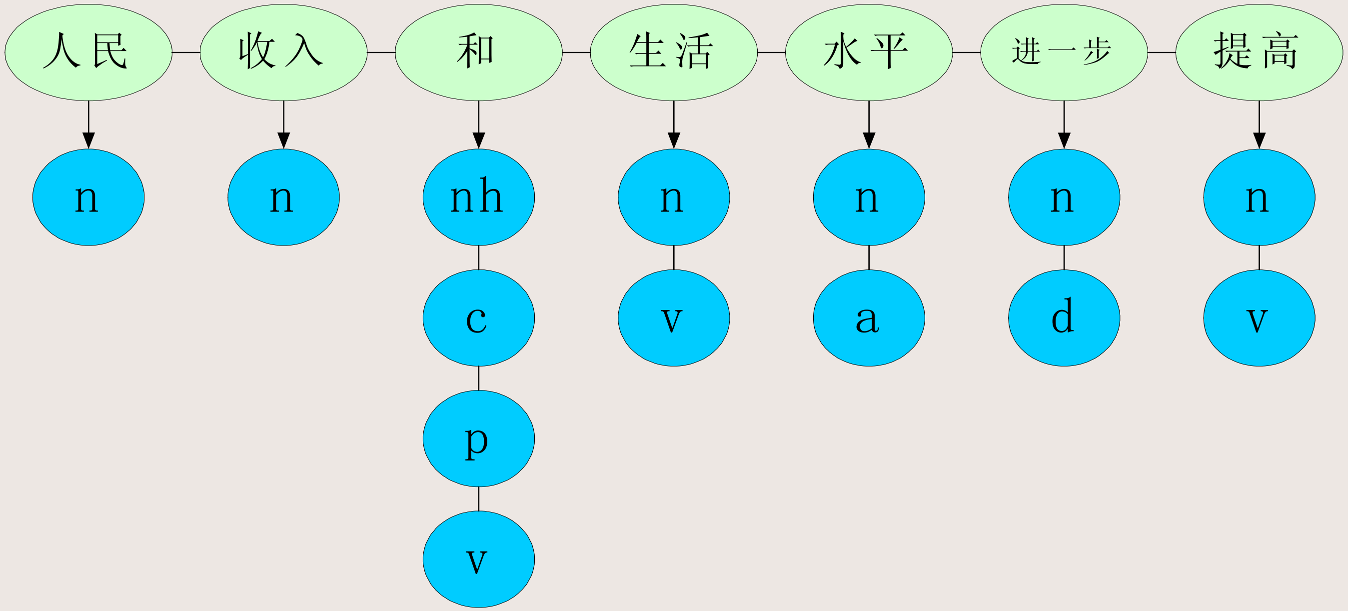
- 语料库(只有两个句子)
  - A lion ran to the rock
  - D N V P D N
  - Aux V
  - The cat slept on the mat
  - D N V P D N
  - V R
- 我们能够学习到什么？
  - D, N, V的概率大于D, V, V，Cat应该标注为N
  - V, P, D的概率大于V, Aux, D或V, R, D，因此to和on应标为P

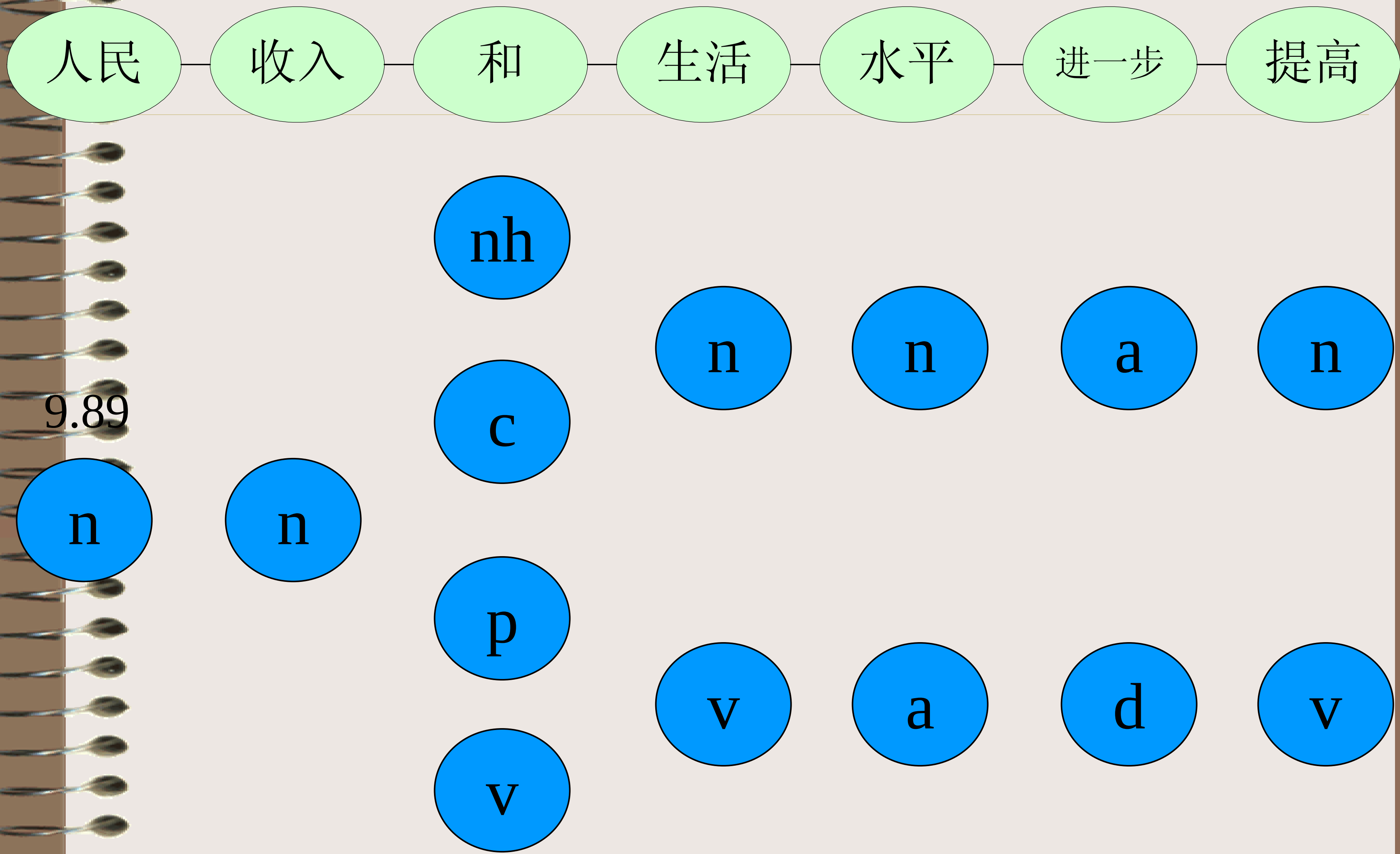
未登录词

- 考虑所有词性
- 只考虑开放类词性
  – Uniform（平均分配概率）
  – Unigram（考虑每个词性独立出现的概率）
- 根据未登录词的前缀和后缀猜测其词性

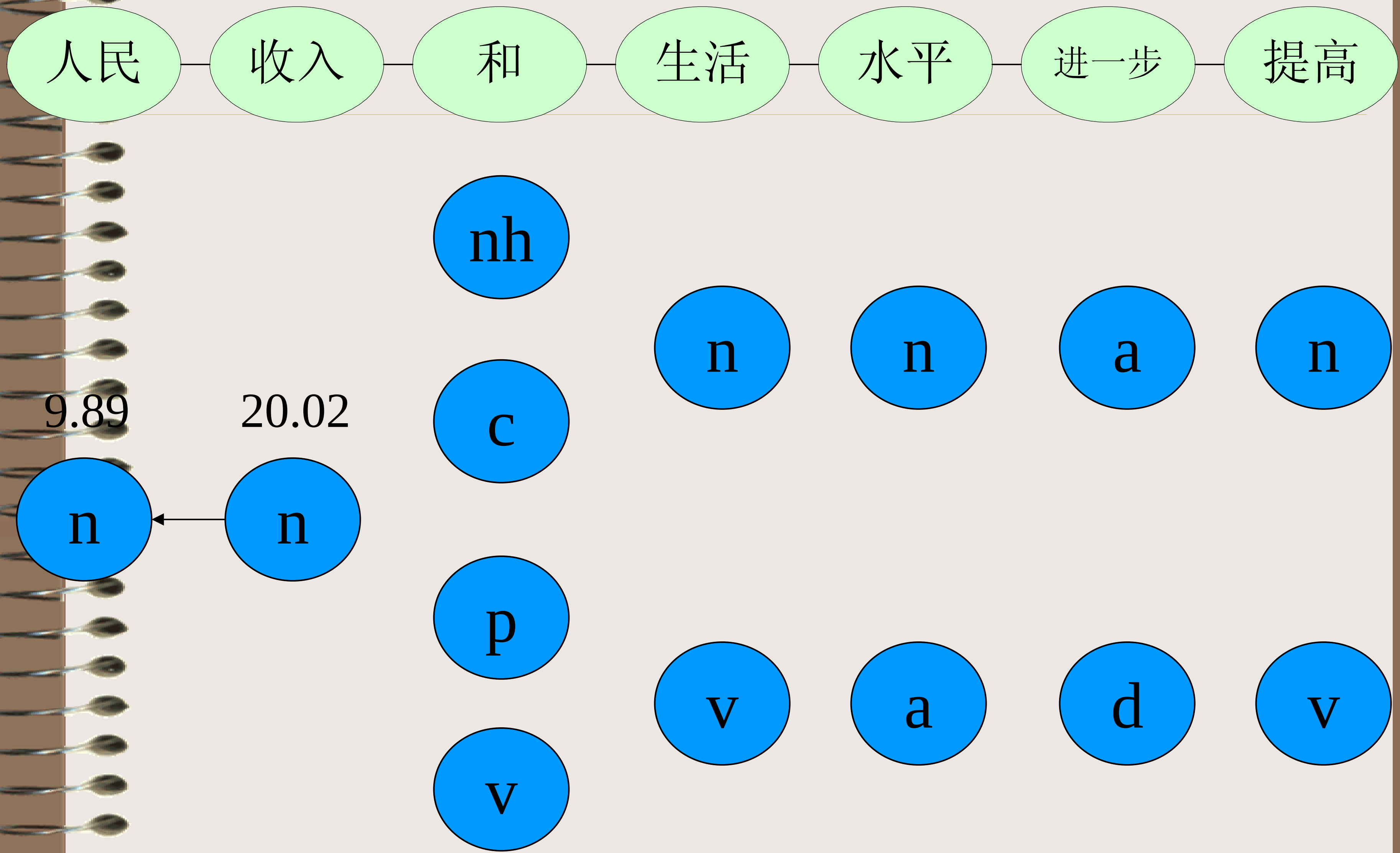
运行词性标注器

无论是对有指导的学习，还是对无指导的学习，在搜索阶段都一样：使用Viterbi算法！

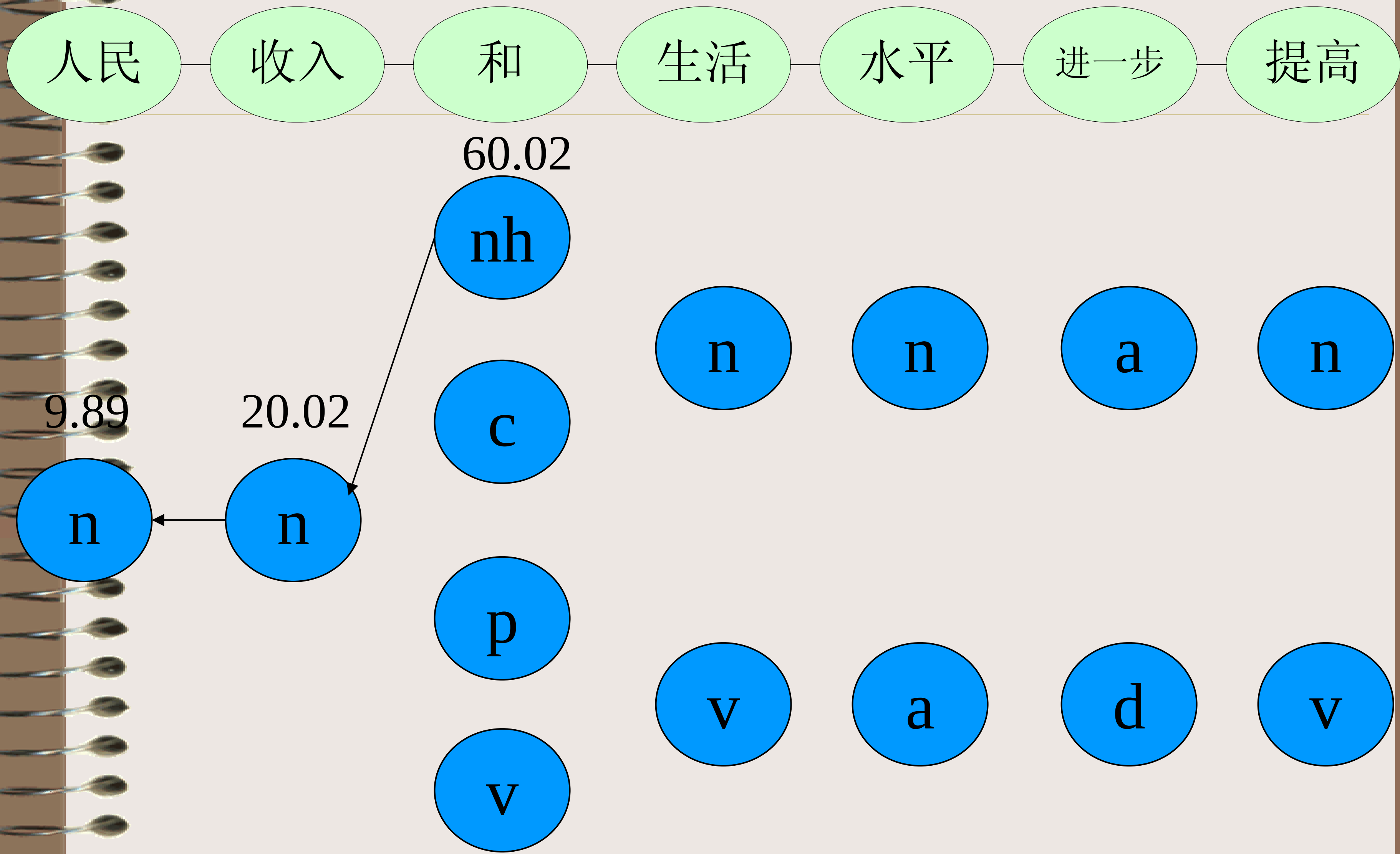




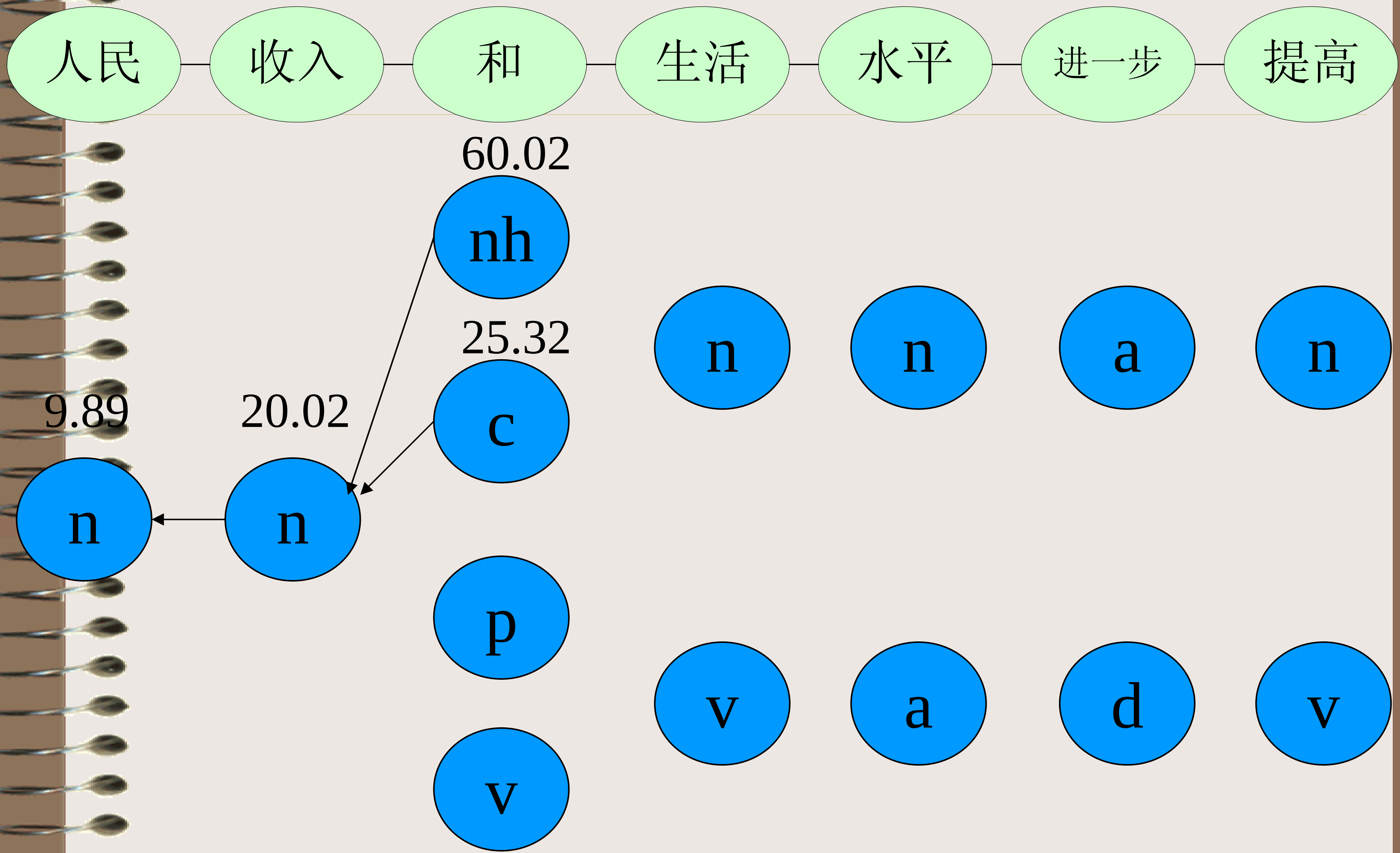
$\Pi_n=2.52$ $b_n(人民)=7.37$



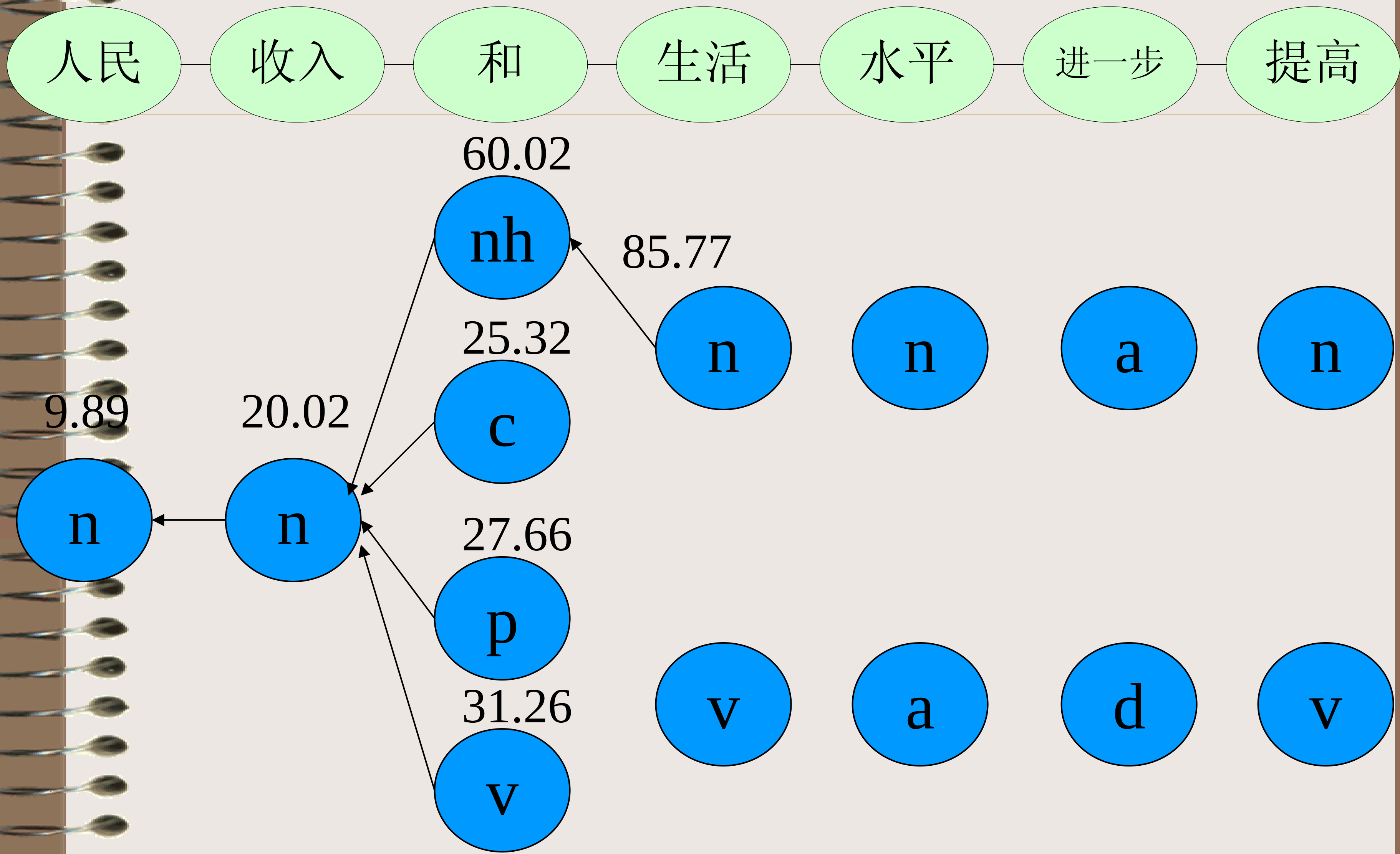
$b_n(\text{收入})=6.98$ $a_{nn}=2.76$



b_{nh}(和)=20 $a_{n\ nh}$=20

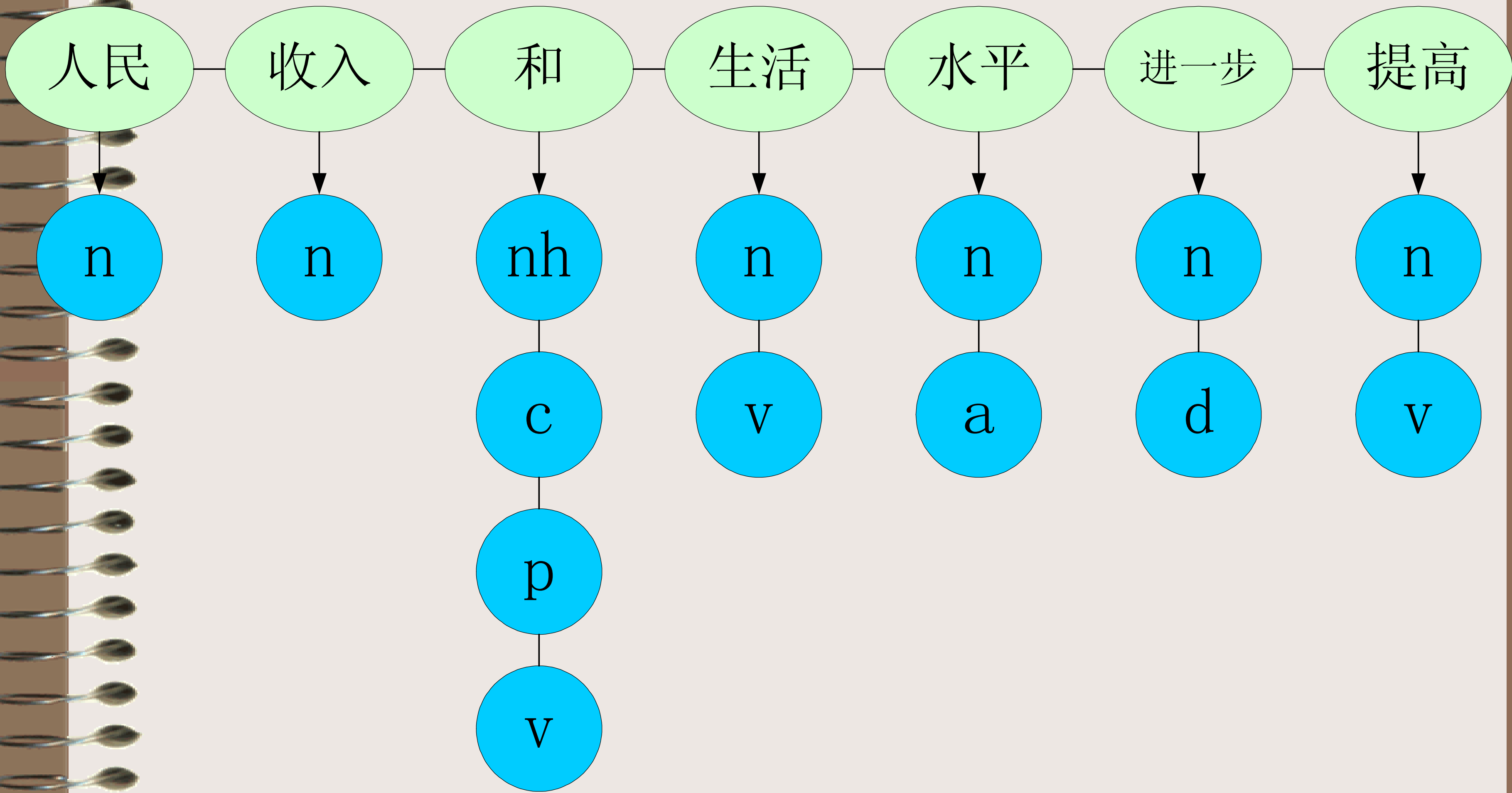


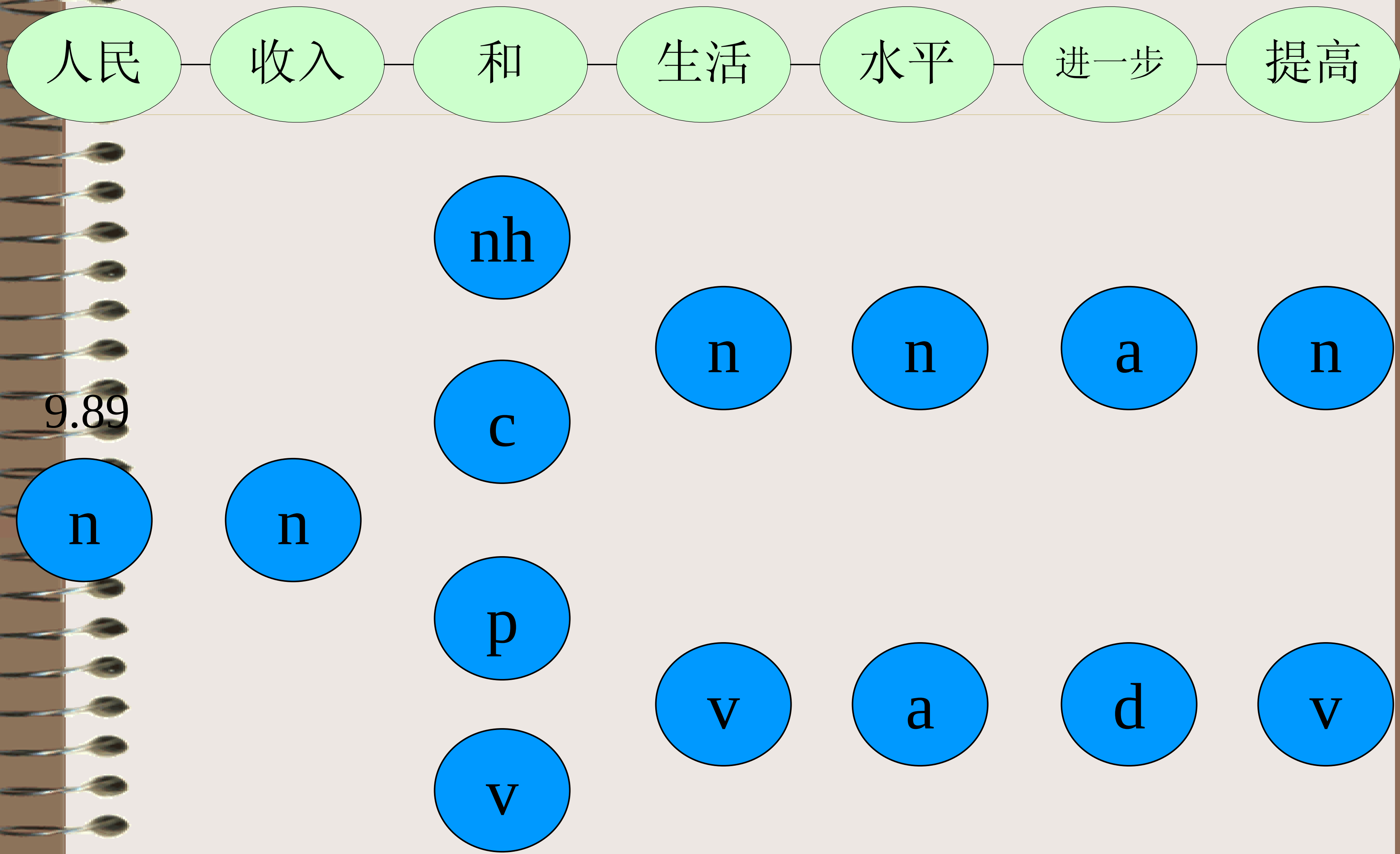
$b_c(和)=1.72$ $a_{n\ c}=3.58$



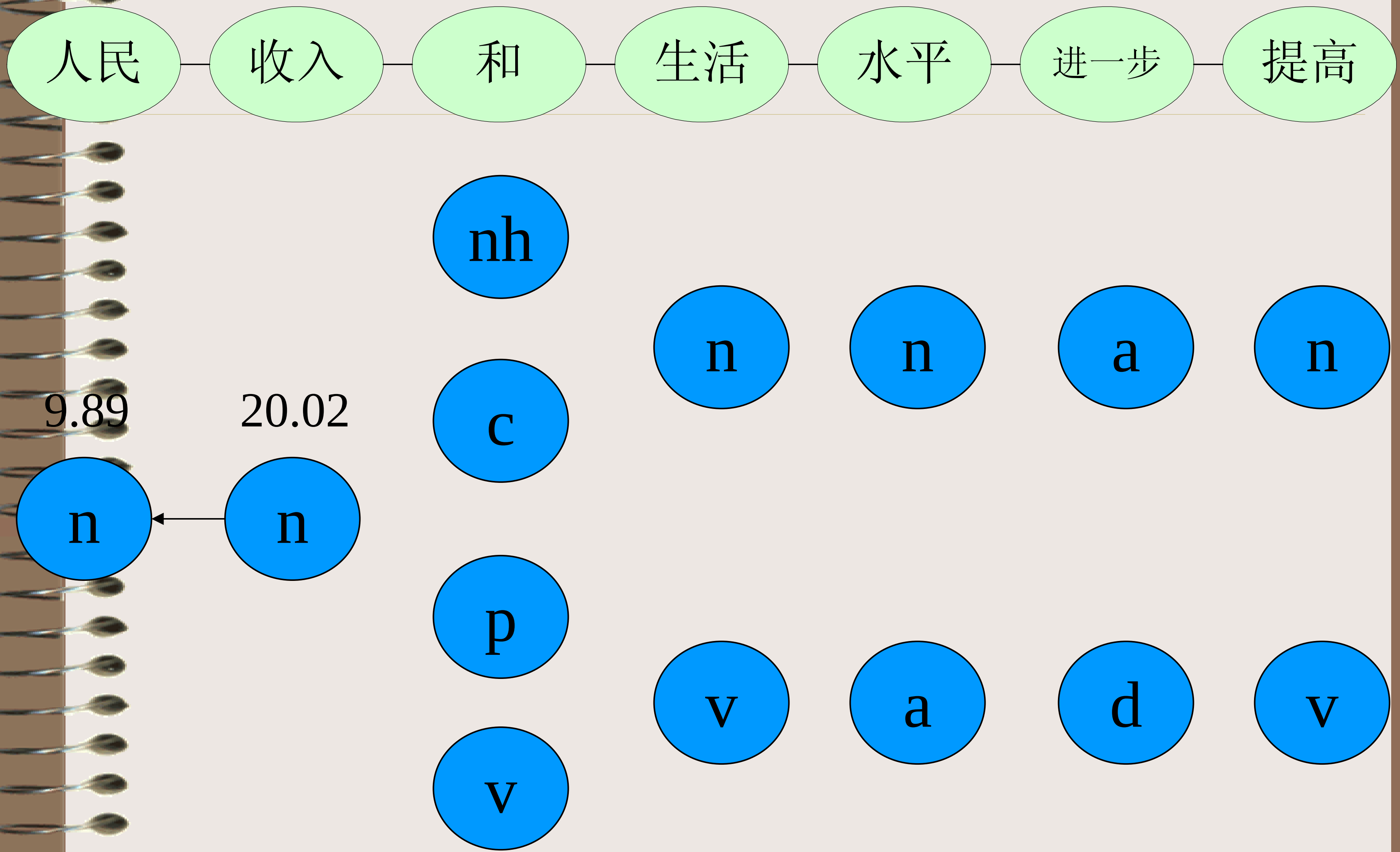
b_n(生活)=5. 75 $a_{nh\ n}$=20

Viterbi算法举例



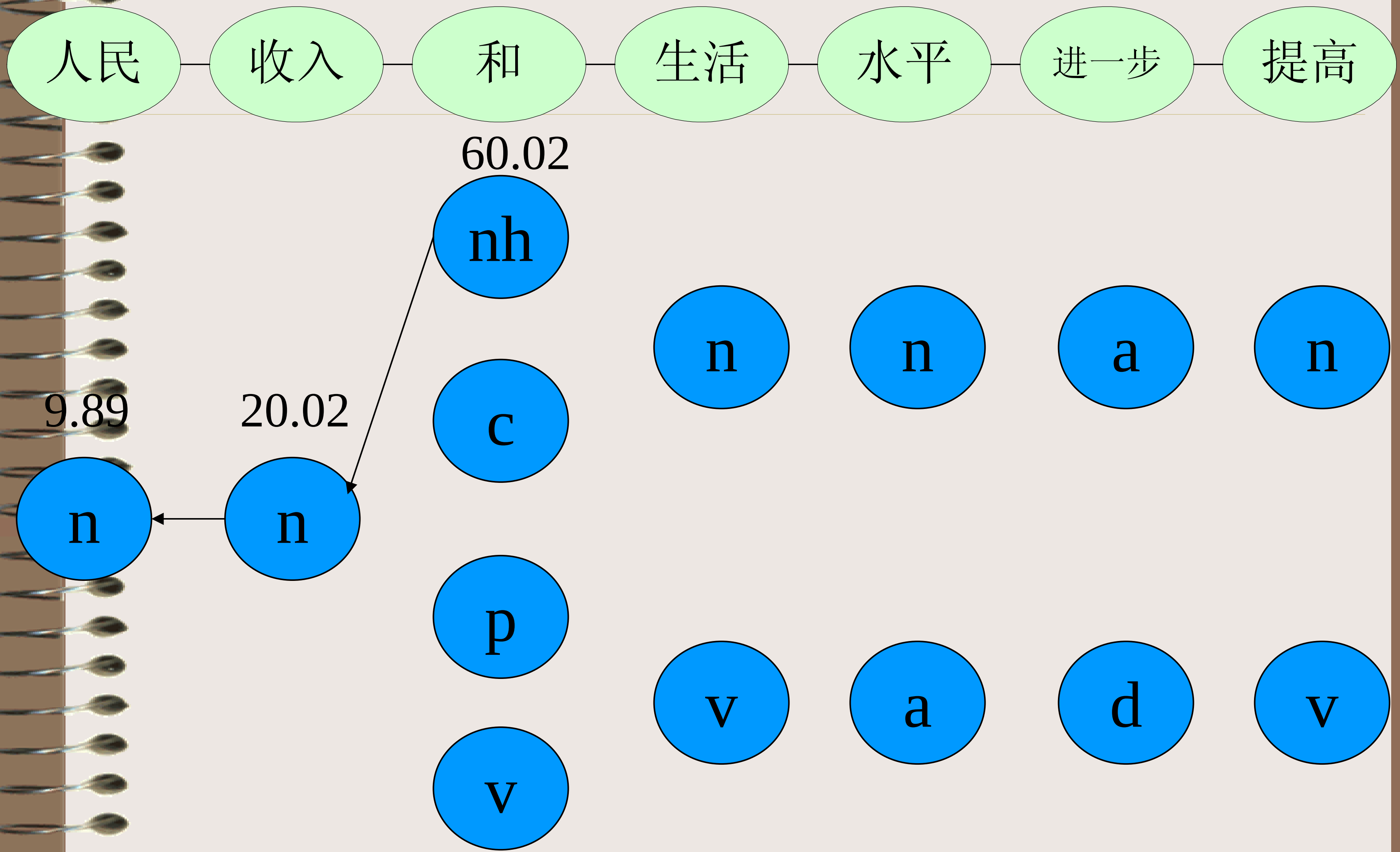


Π_n=2. 52 b_n(人民)=7. 37

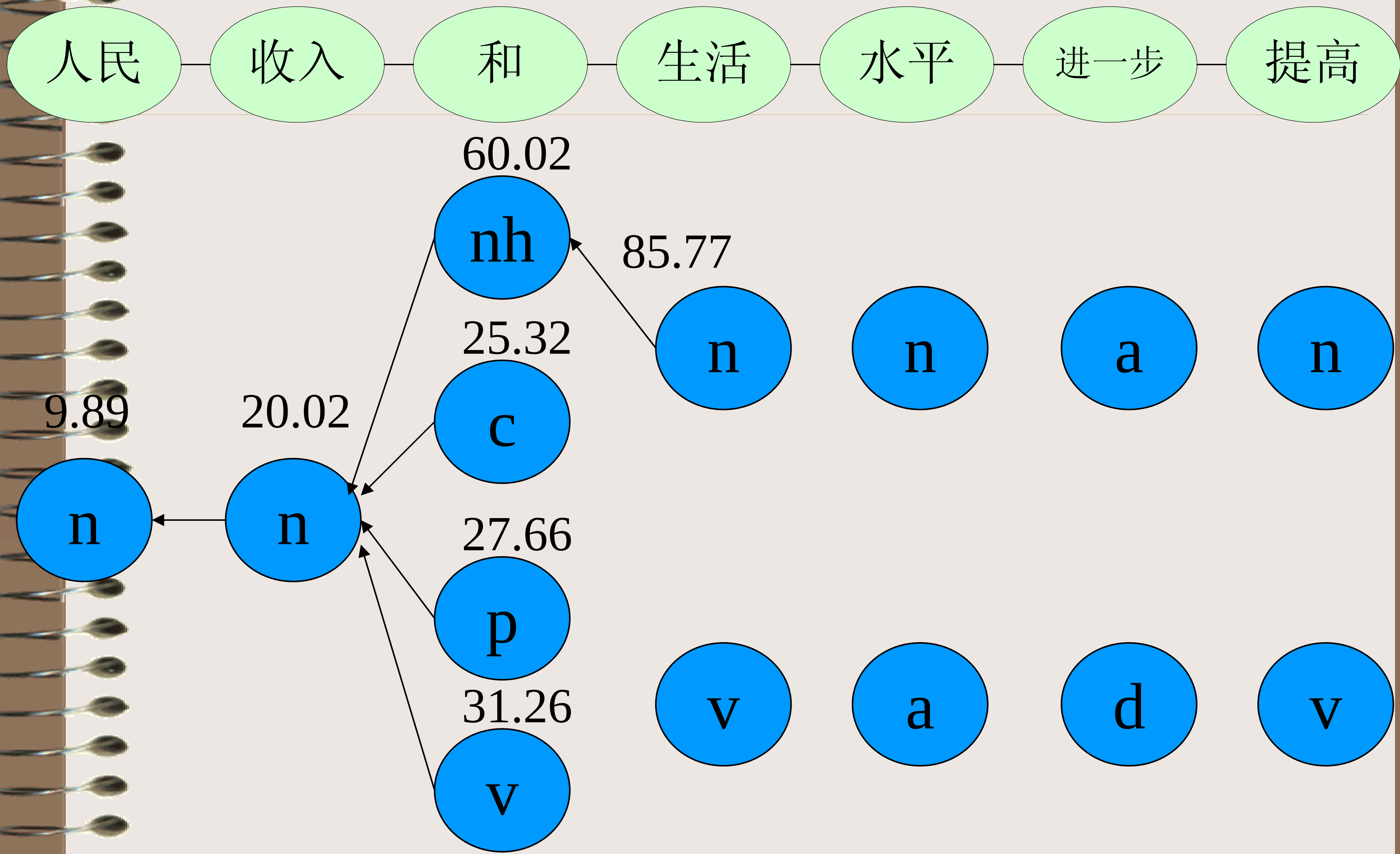


b_n(收入)=6. 98

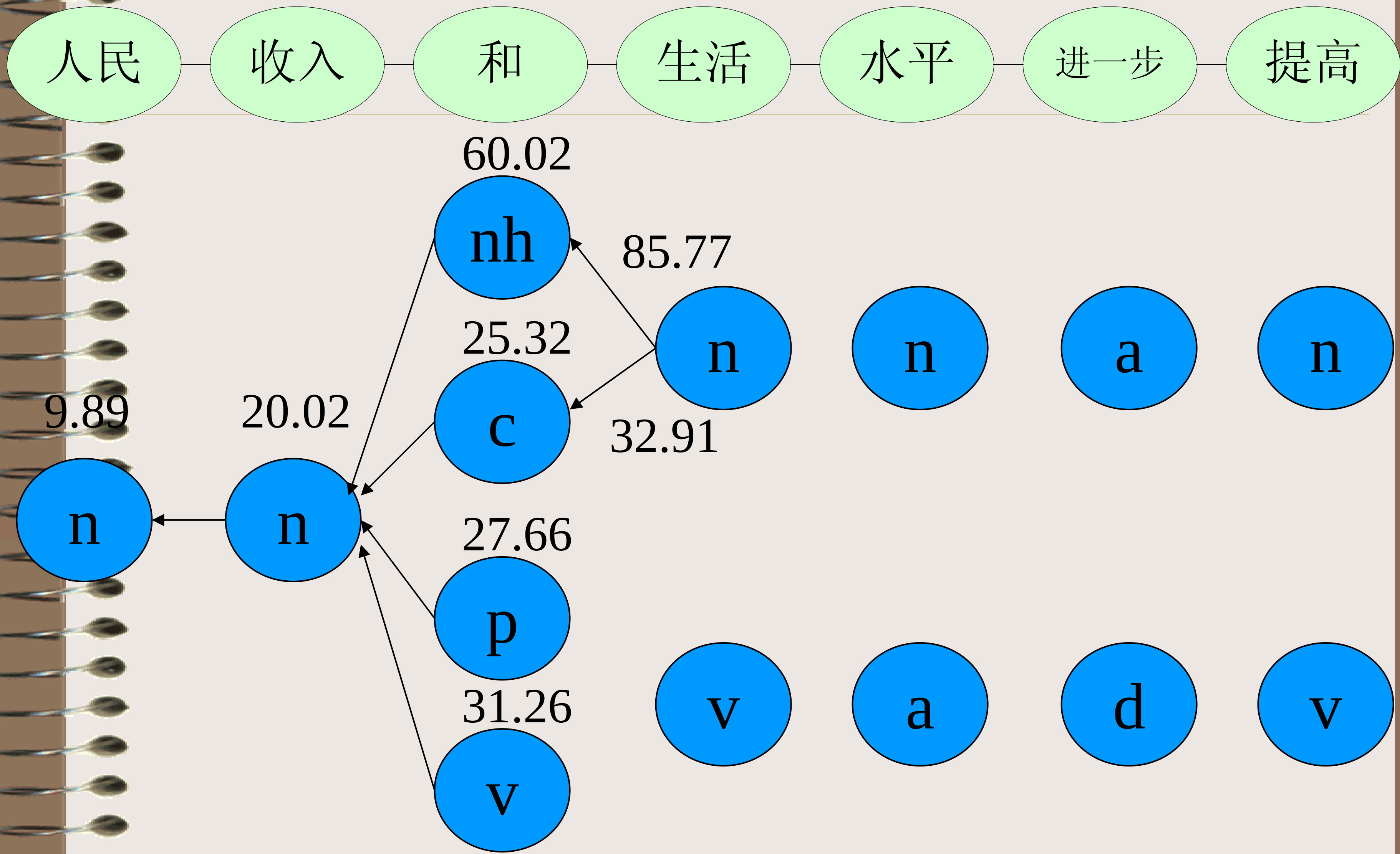
a_{nn}=2. 76



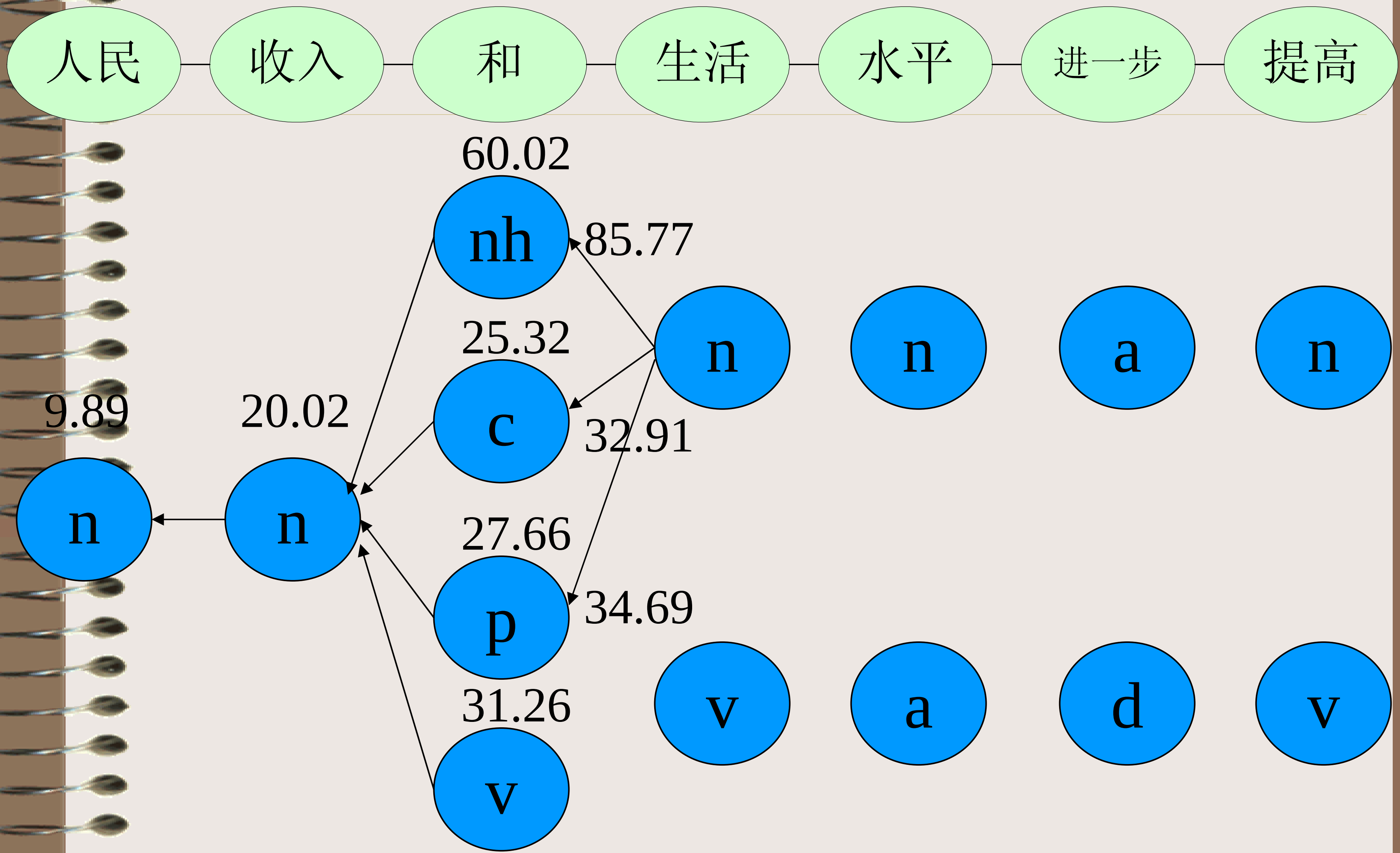
b_{nh}(和)=20 $a_{n\ nh}$=20



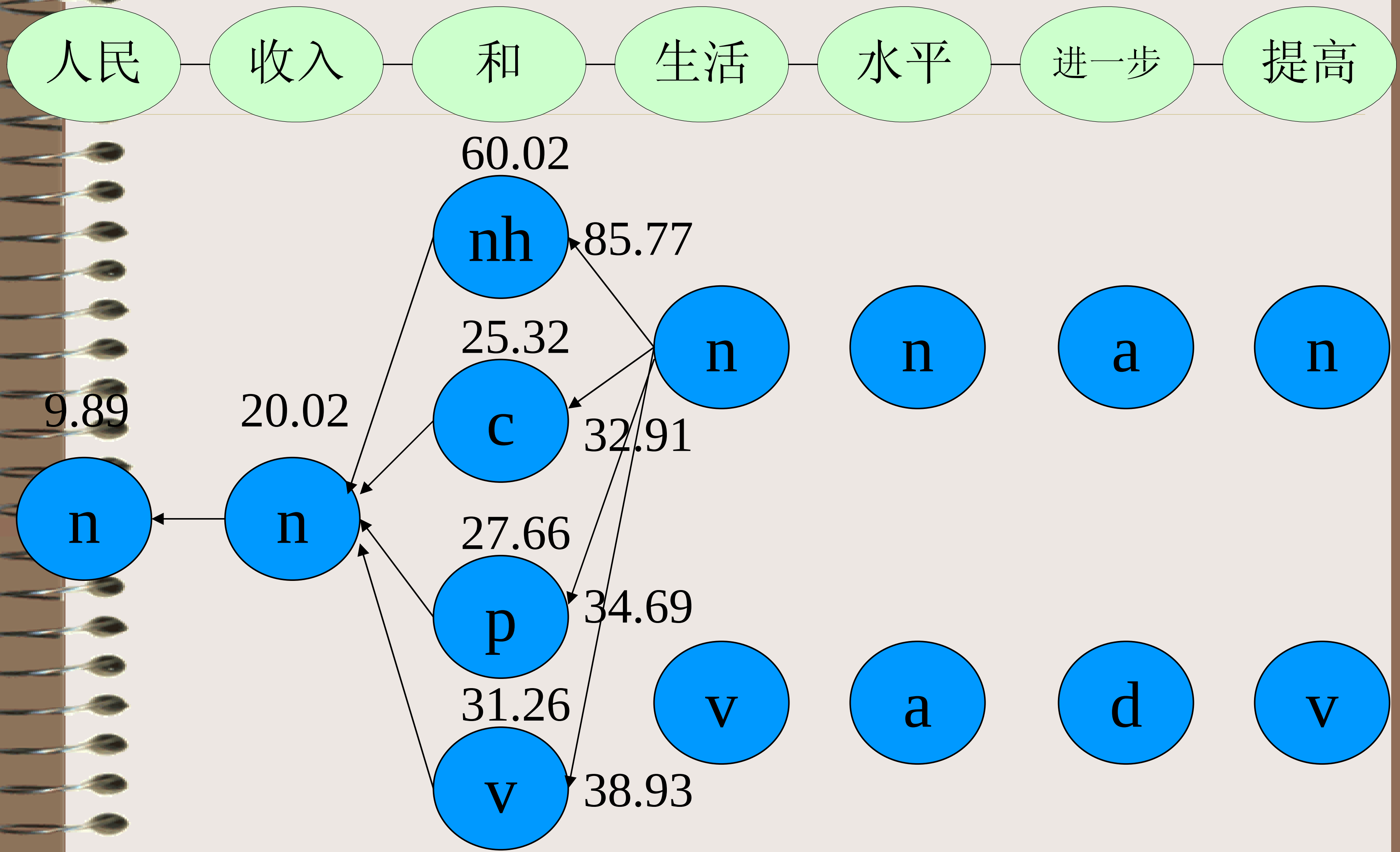
b_n(生活)=5. 75 $a_{nh\ n}$=20



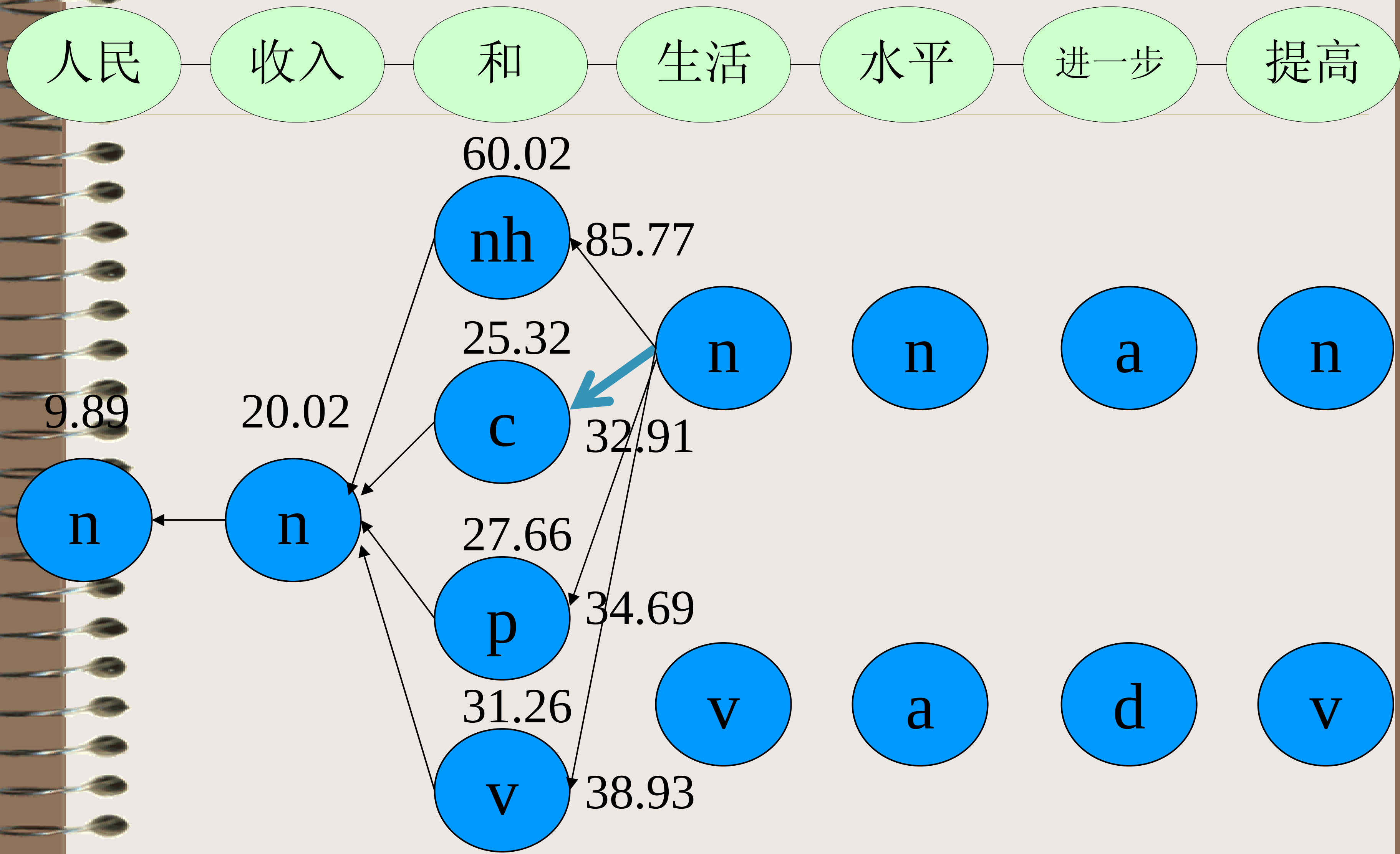
b_n(生活)=5. 75　　$a_{c\ n}$=1. 84



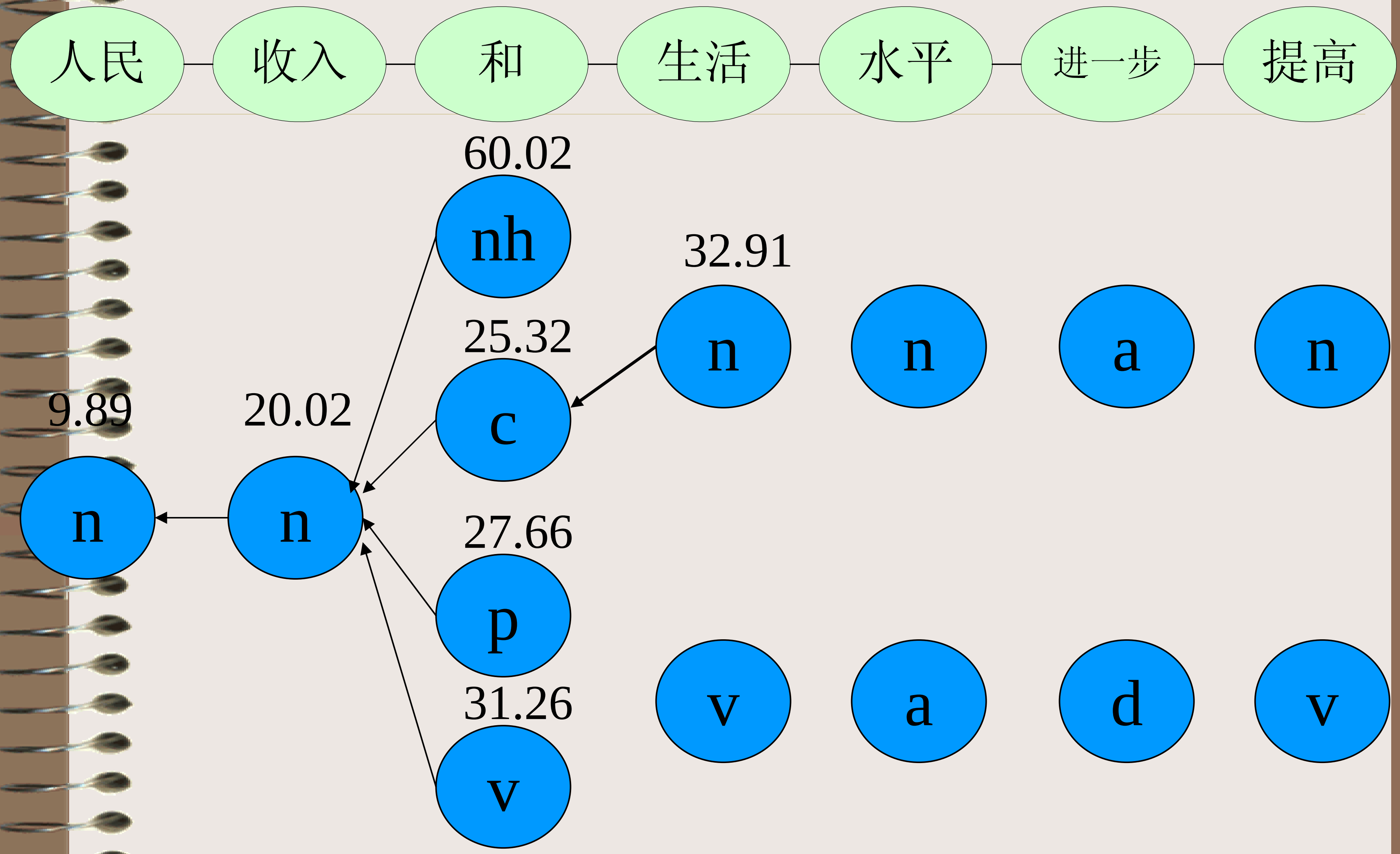
$b_n(生活)=5.75$　　$a_{p\ n}=1.28$



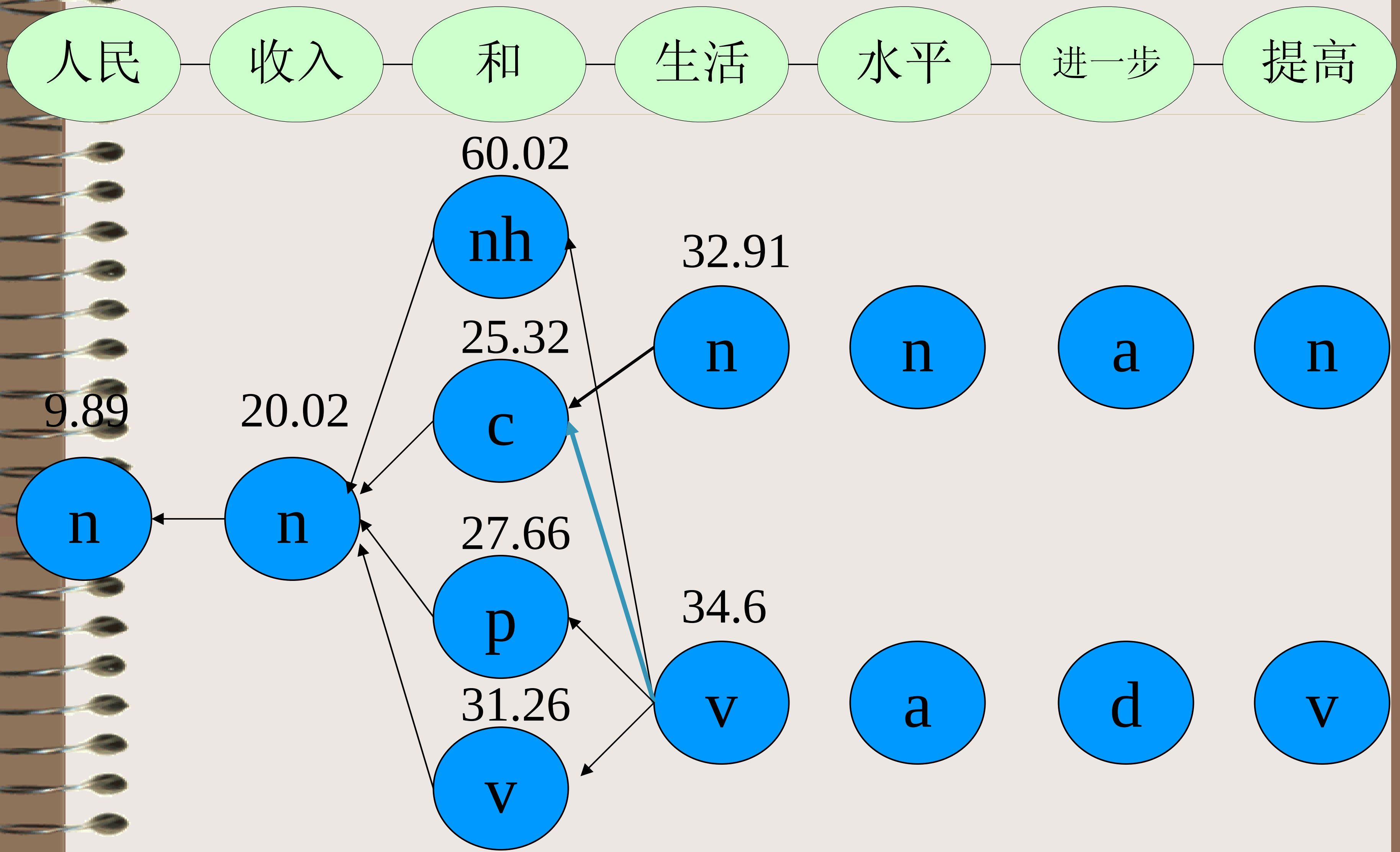
b_n(生活)=5.75　$a_{v\ n}$=1.92



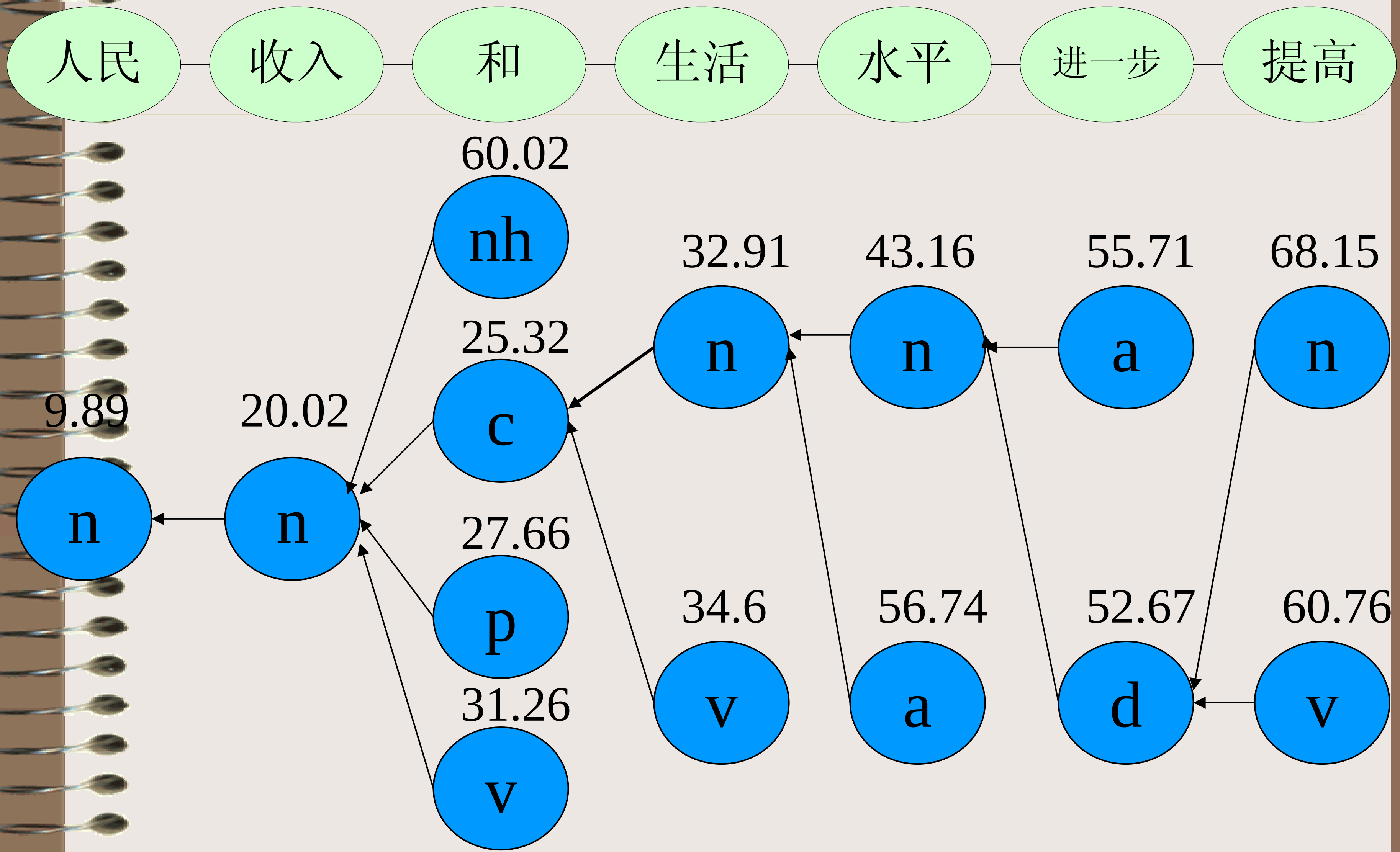
人民
收入
和
生活
水平
进一步
提高
60.02
nh
85.77
25.32
c
32.91
9.89
20.02
n
n
n
n
a
n
27.66
p
34.69
31.26
v
38.93
v
a
d
v

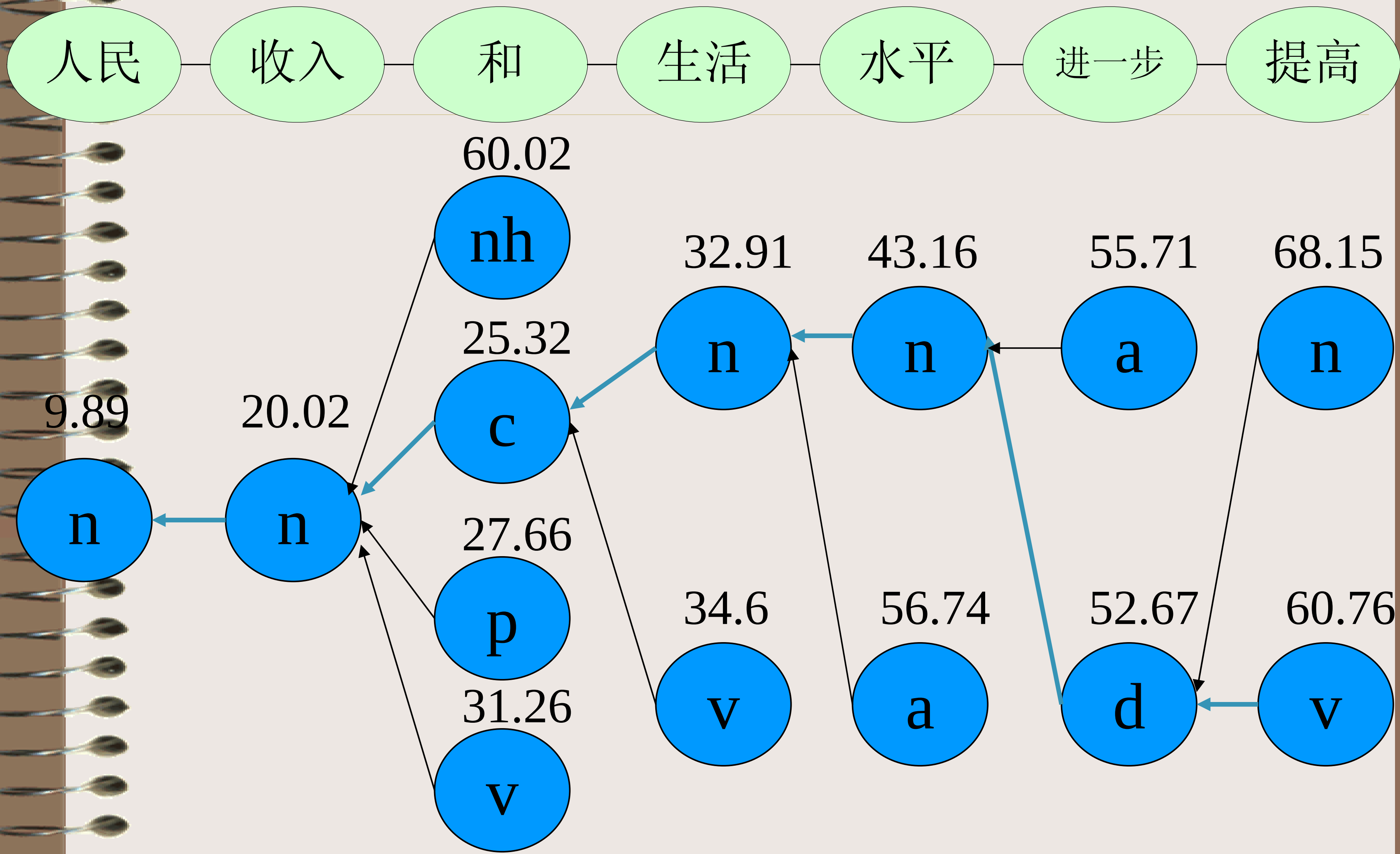


人民
收入
和
生活
水平
进一步
提高
60.02
nh
32.91
25.32
n
n
a
n
9.89
20.02
c
n
n
27.66
p
31.26
v
a
d
v
v

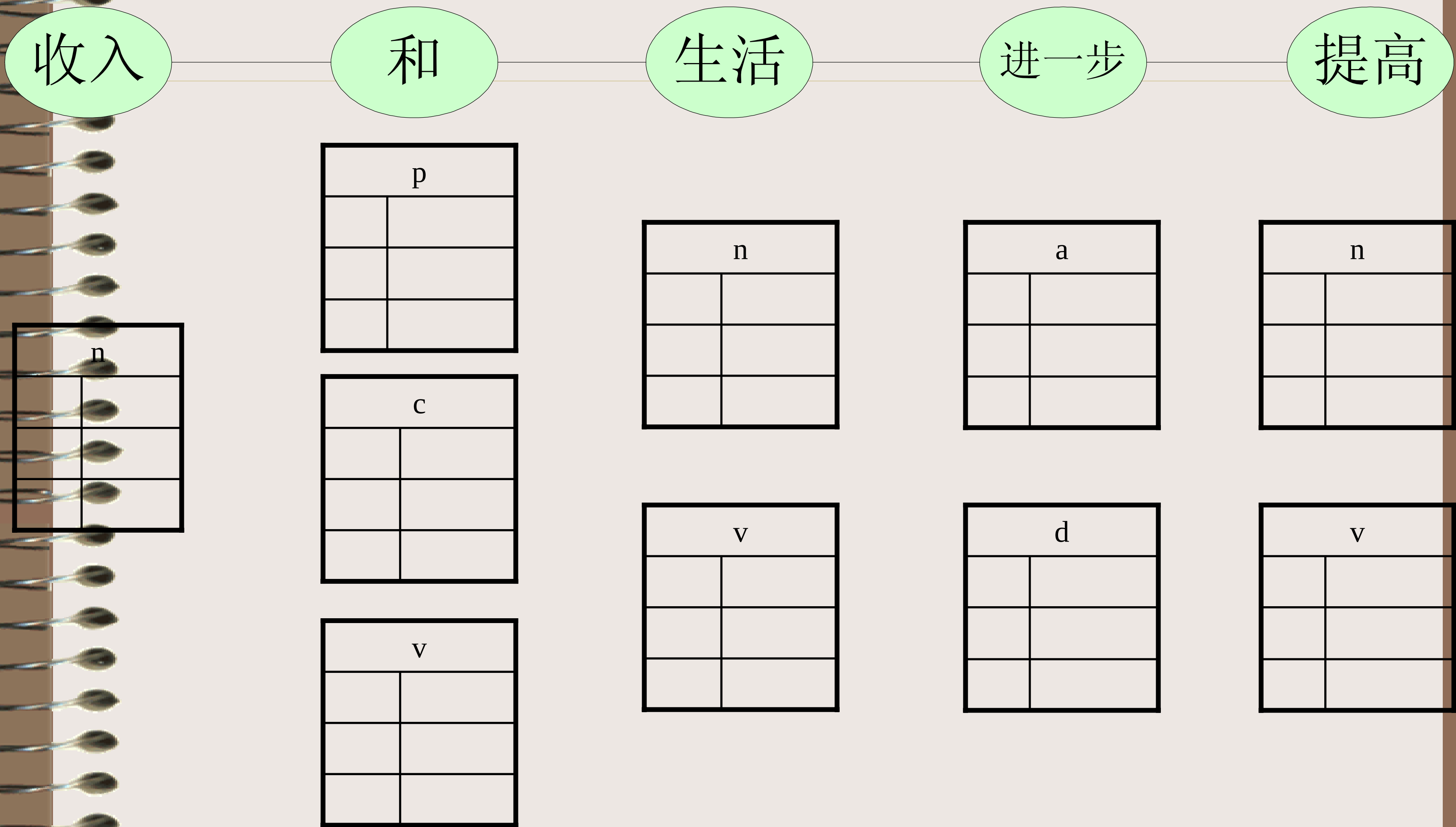


人民
收入
和
生活
水平
进一步
提高
60.02
nh
32.91
25.32
n
n
a
n
9.89
20.02
c
n
n
27.66
p
34.6
31.26
v
a
d
v
v


人民
收入
和
生活
水平
进一步
提高
60.02
nh
32.91
43.16
55.71
68.15
25.32
n
n
a
n
9.89
20.02
c
n
n
27.66
34.6
56.74
52.67
60.76
p
31.26
v
a
d
v
v

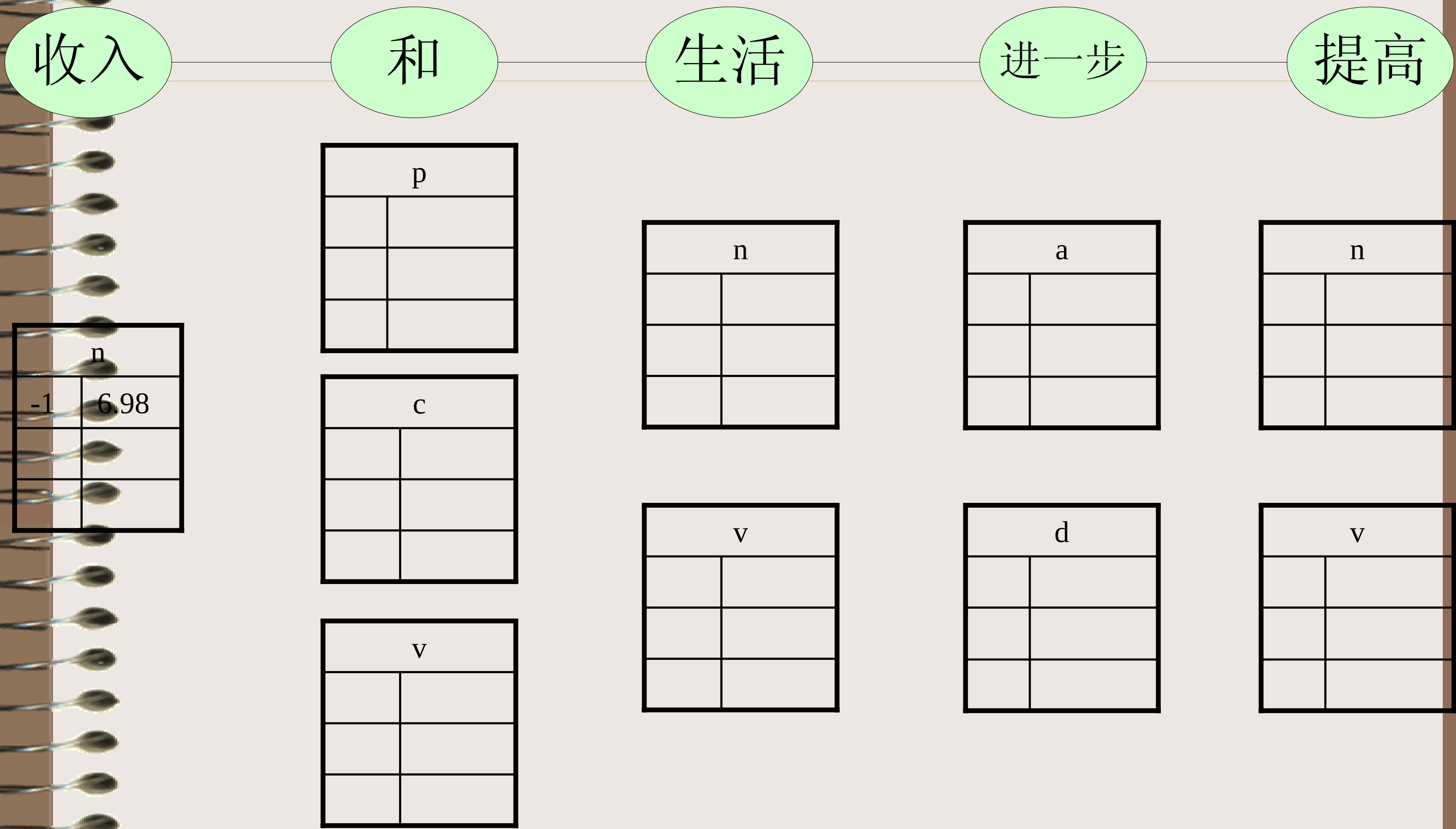


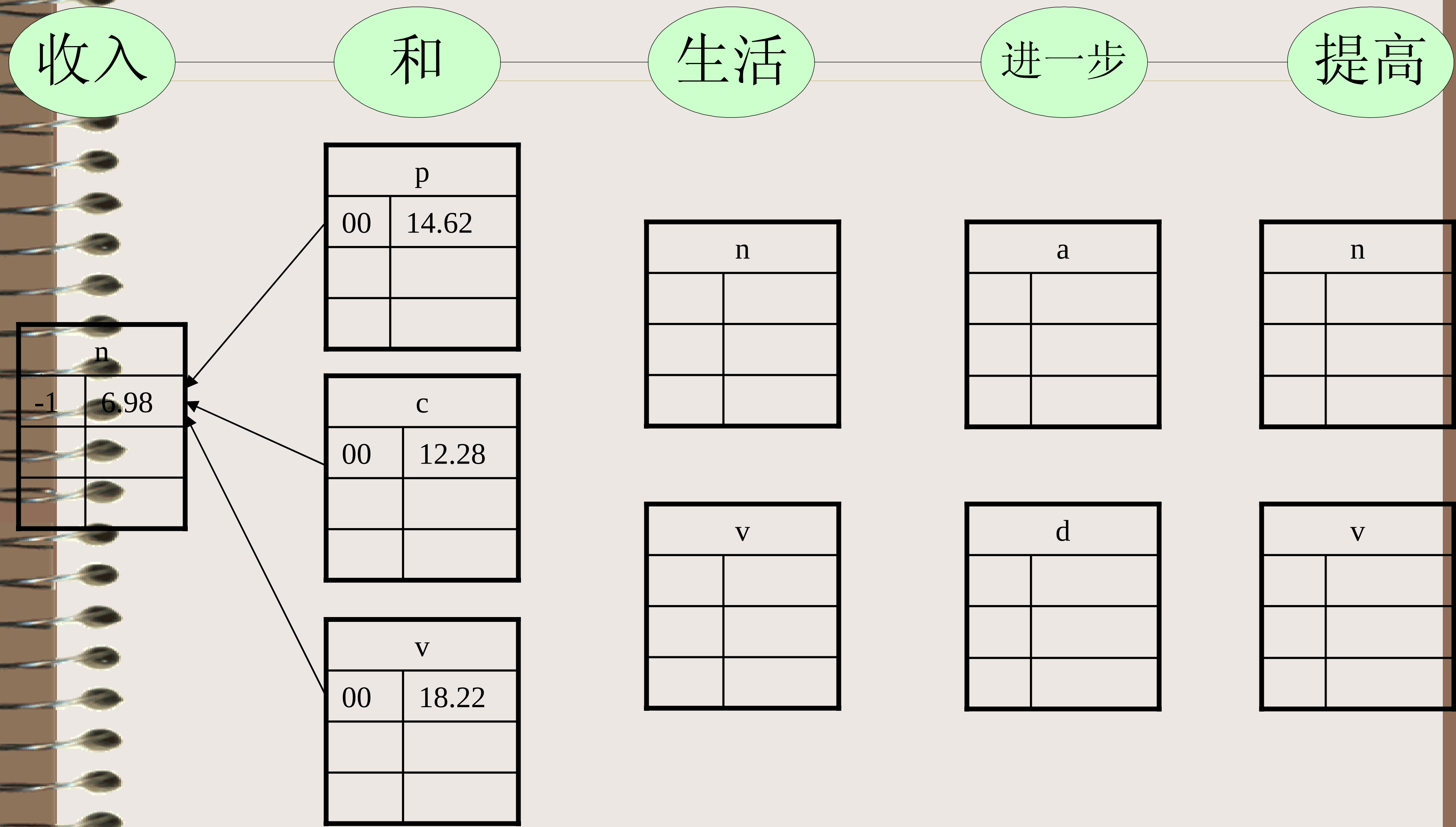
人民/n 收入/n 和/c 生活/n 水平/n 进一步/d 提高/v



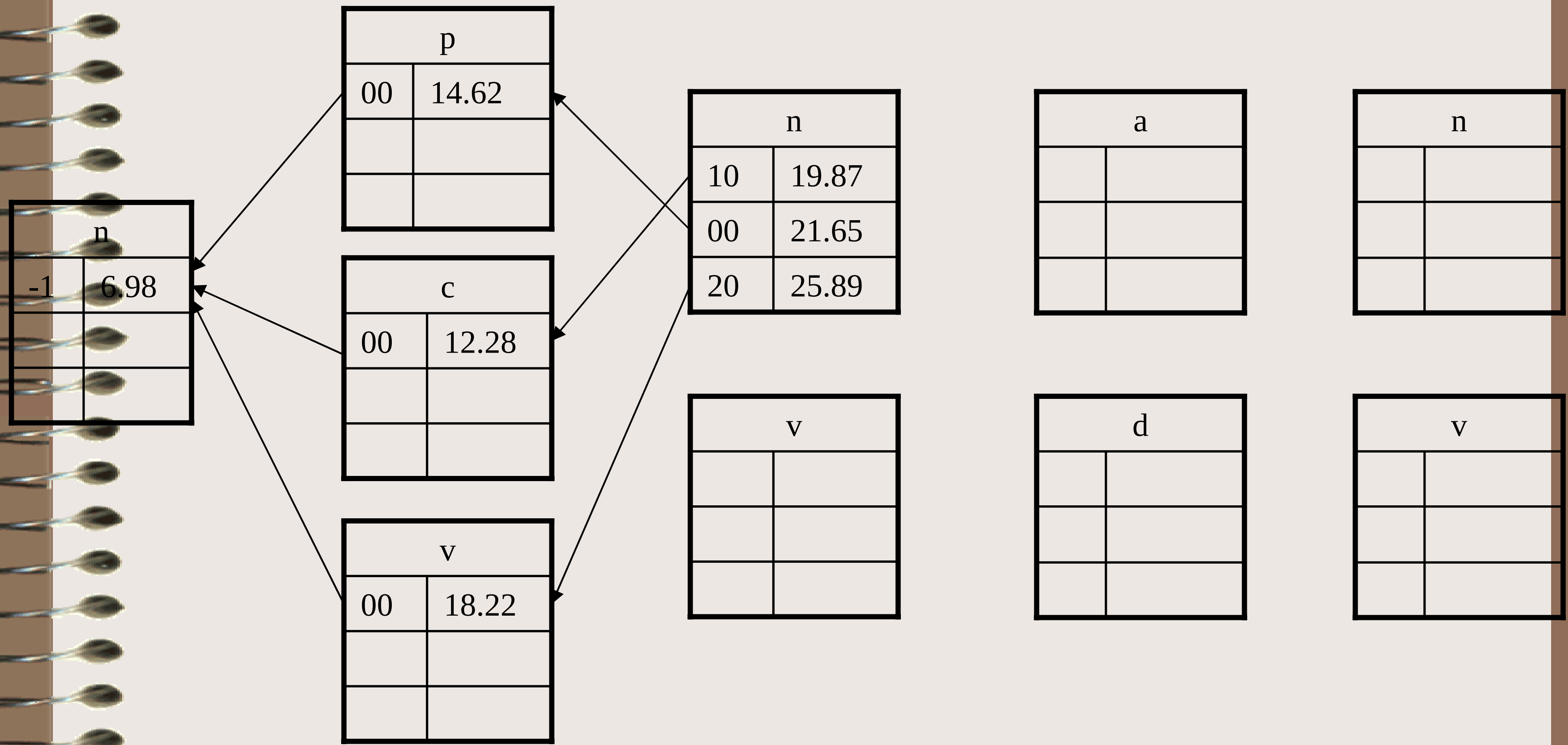
收入
和
生活
进一步
提高
p
n
c
a
n
n
v
v
d
v

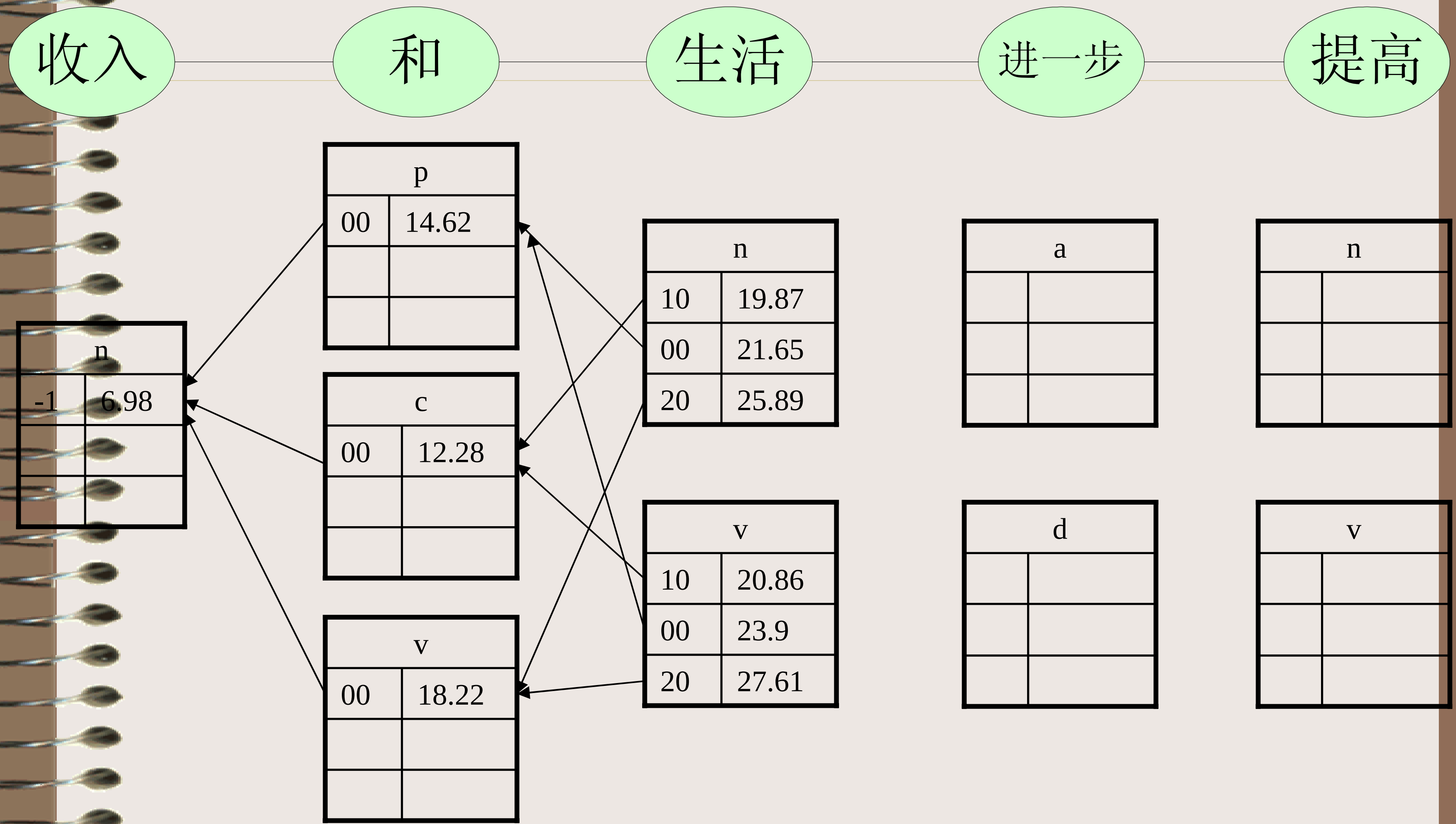
N-Best结果

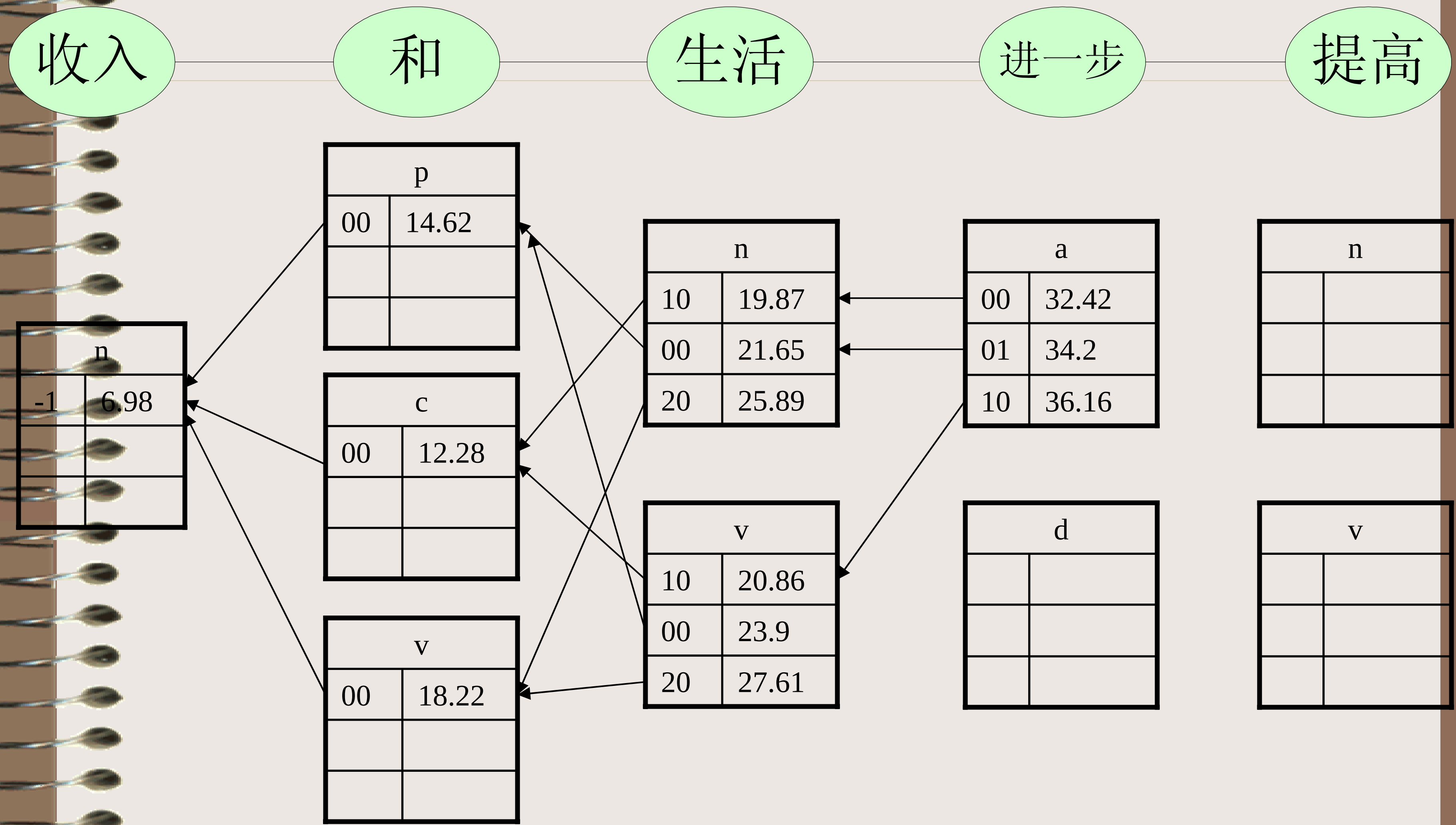



收入
和
生活
进一步
提高
p
00
14.62
n
-1
6.98
c
00
12.28
v
00
18.22
n
a
n
v
d
v

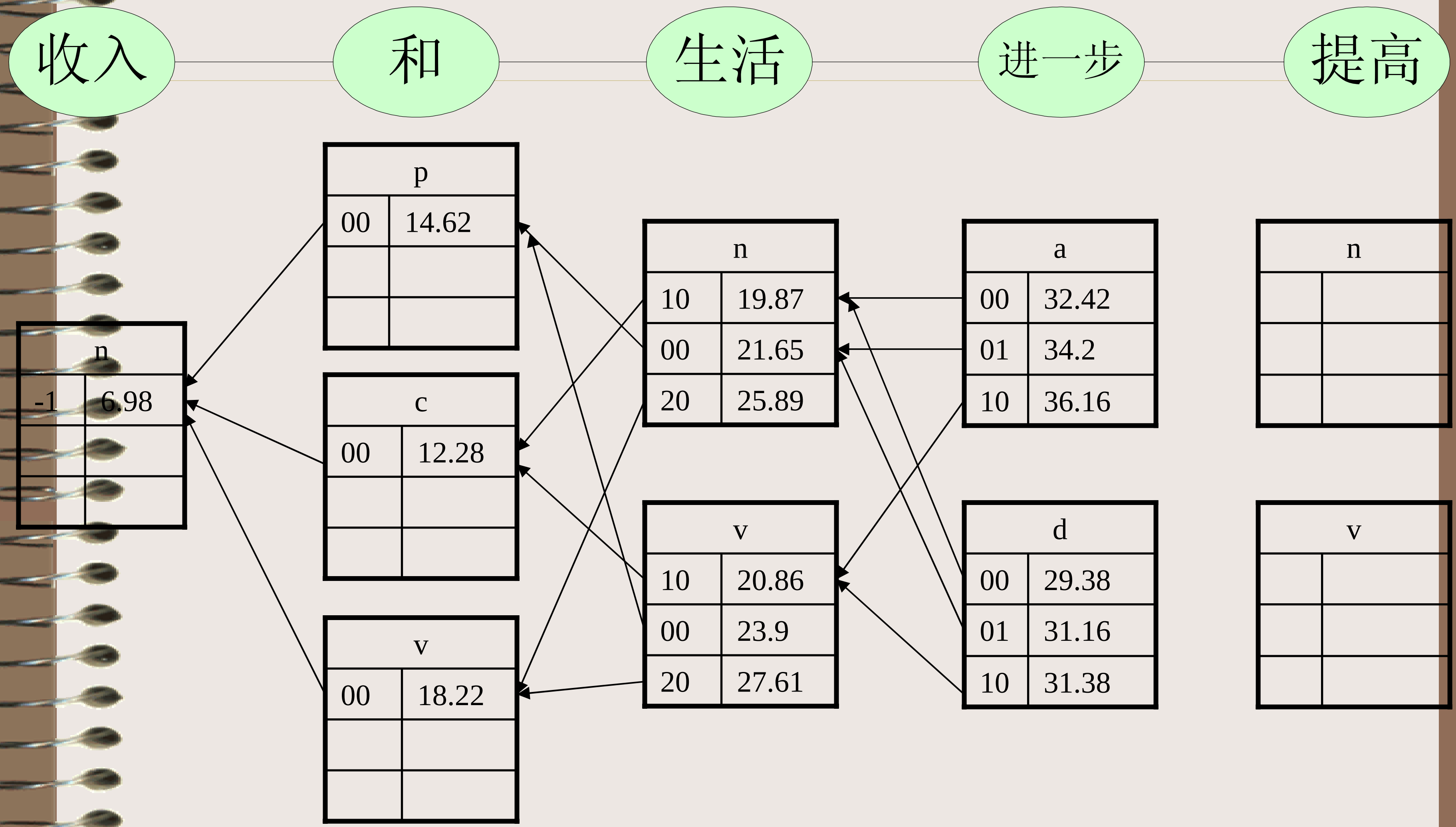
收入 和 生活 进一步 提高


p
00 14.62
n
10 19.87
00 21.65
20 25.89
a
n
n
-1 6.98
c
00 12.28
v
00 18.22
v
d
v

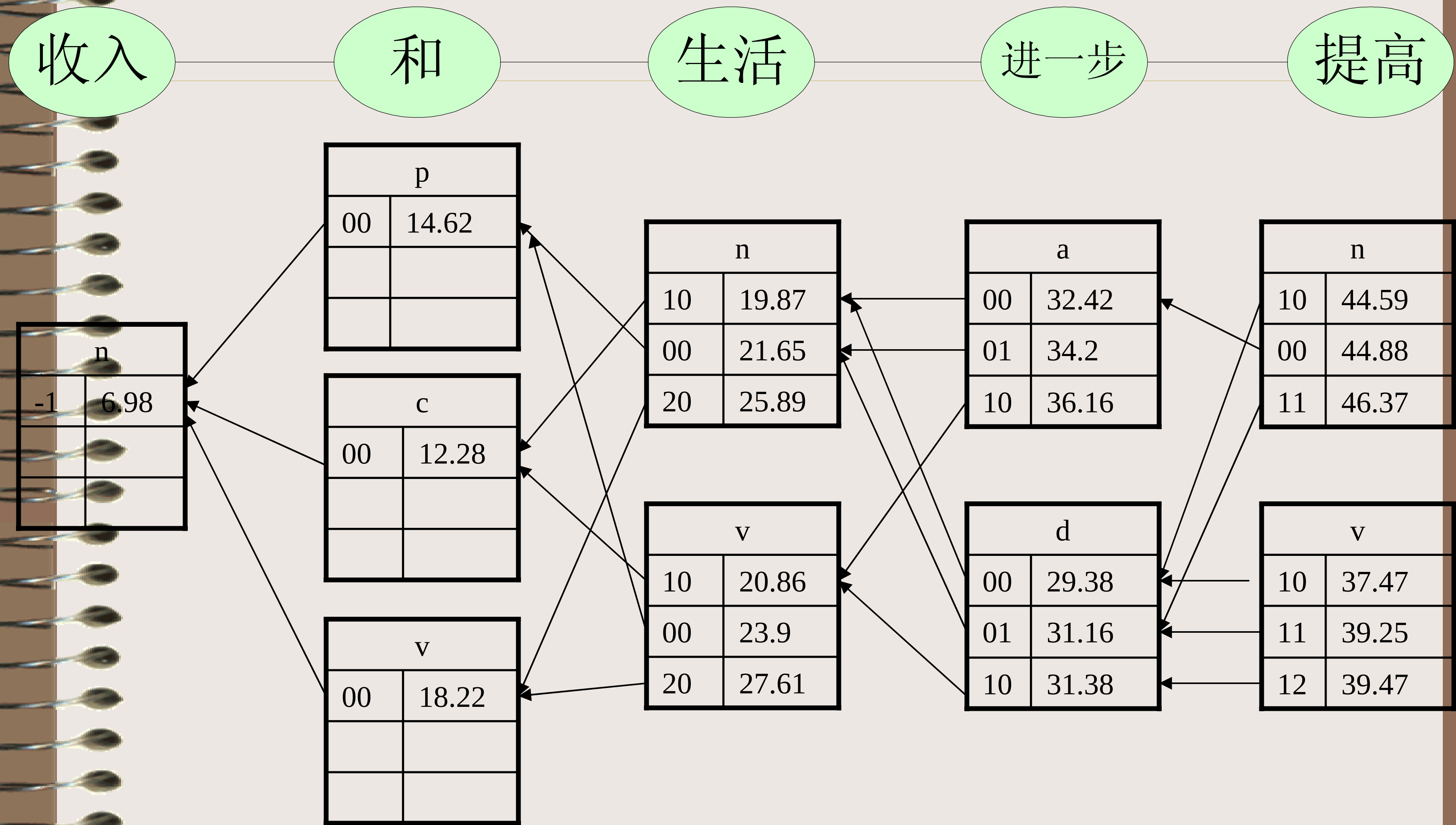

收入
和
生活
进一步
提高
n
-1
6.98
p
00
14.62
c
00
12.28
v
00
18.22
n
10
19.87
00
21.65
20
25.89
v
10
20.86
00
23.9
20
27.61
a
d
n
v



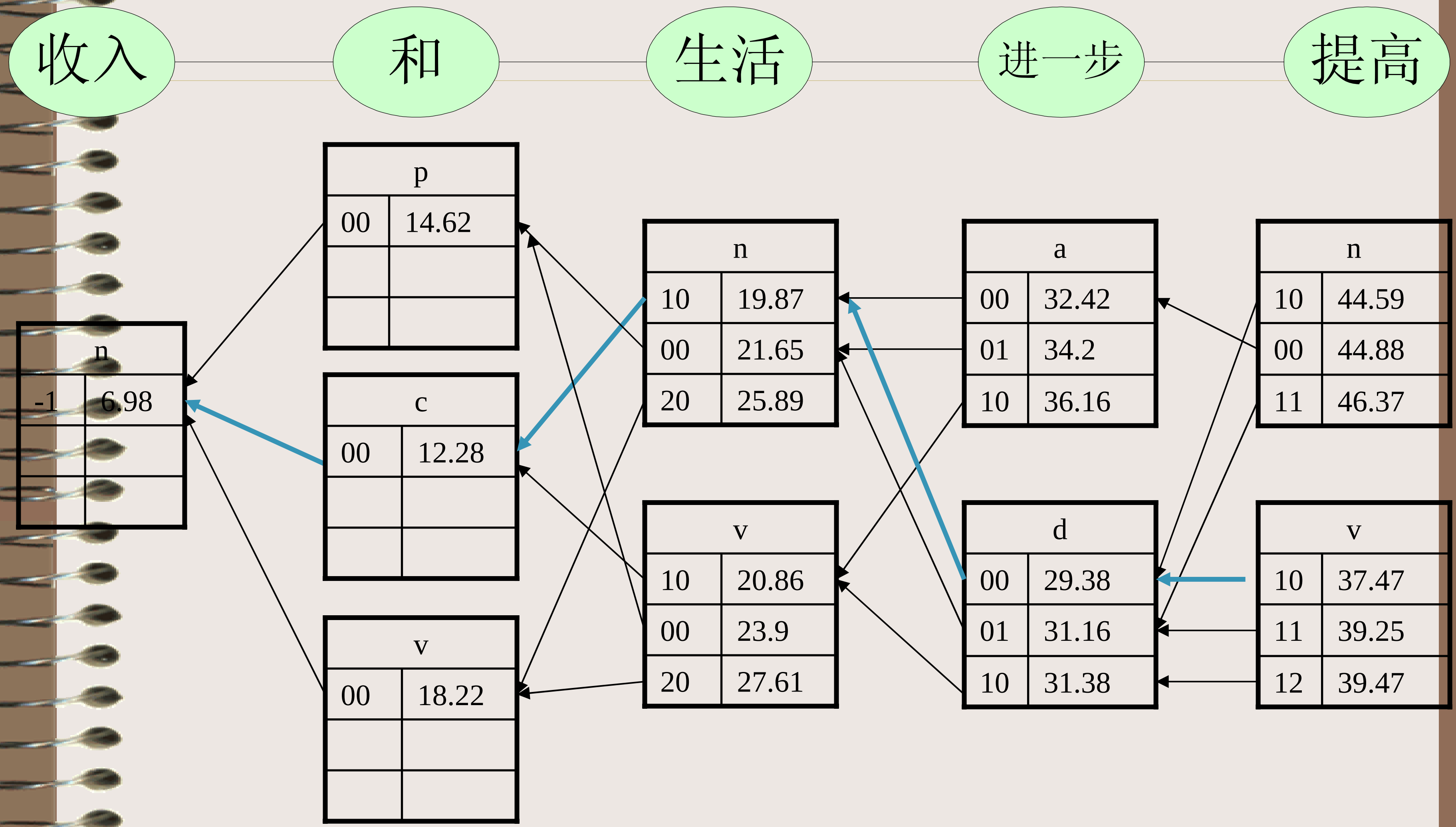
收入
和
生活
进一步
提高
n
-1 6.98
p
00 14.62
c
00 12.28
v
00 18.22
n
10 19.87
00 21.65
20 25.89
v
10 20.86
00 23.9
20 27.61
a
00 32.42
01 34.2
10 36.16
d
n
v

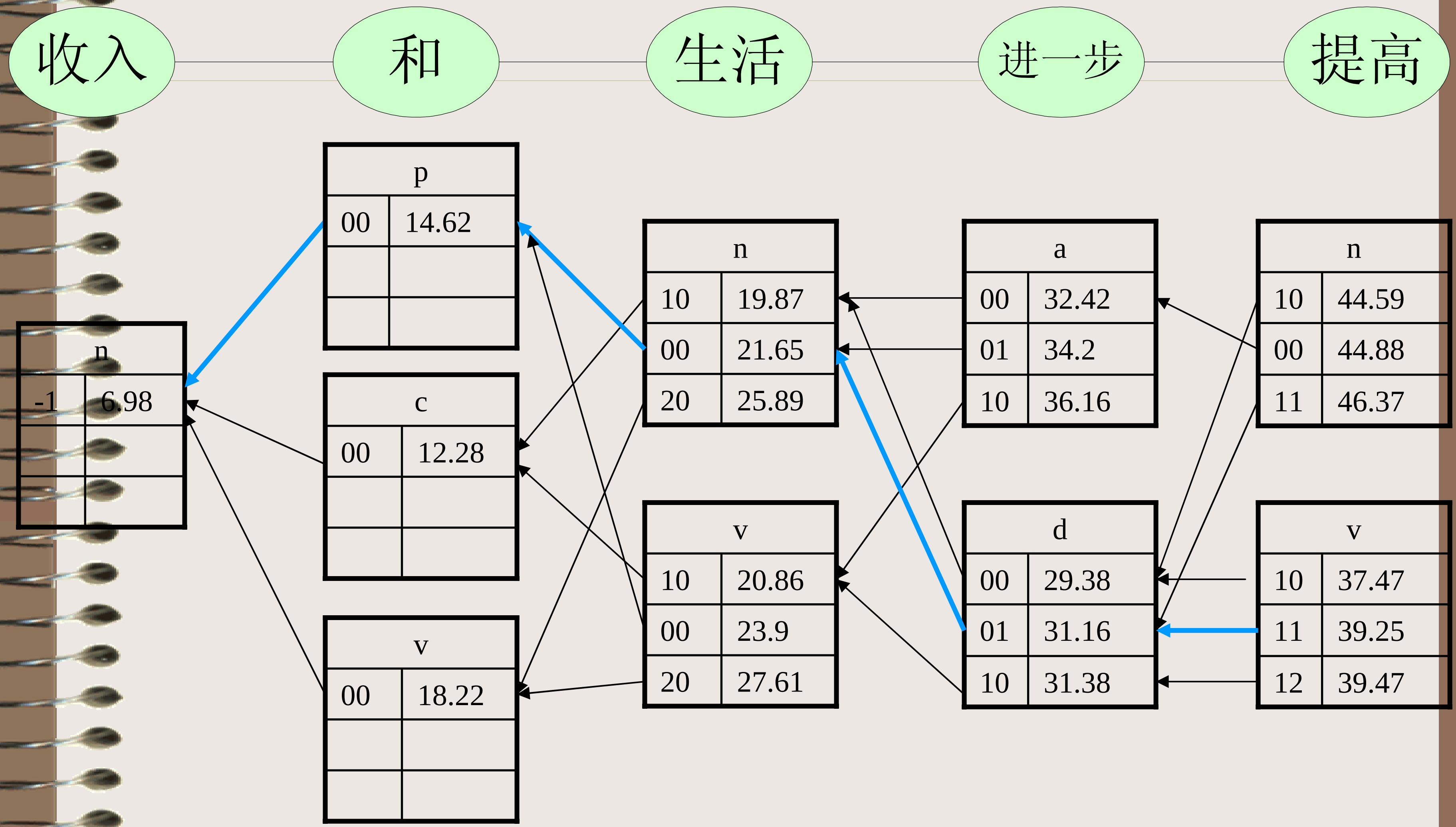


收入
和
生活
进一步
提高
n
-1
6.98
p
00
14.62
c
00
12.28
v
00
18.22
n
10
19.87
00
21.65
20
25.89
v
10
20.86
00
23.9
20
27.61
a
00
32.42
01
34.2
10
36.16
d
00
29.38
01
31.16
10
31.38
n
v

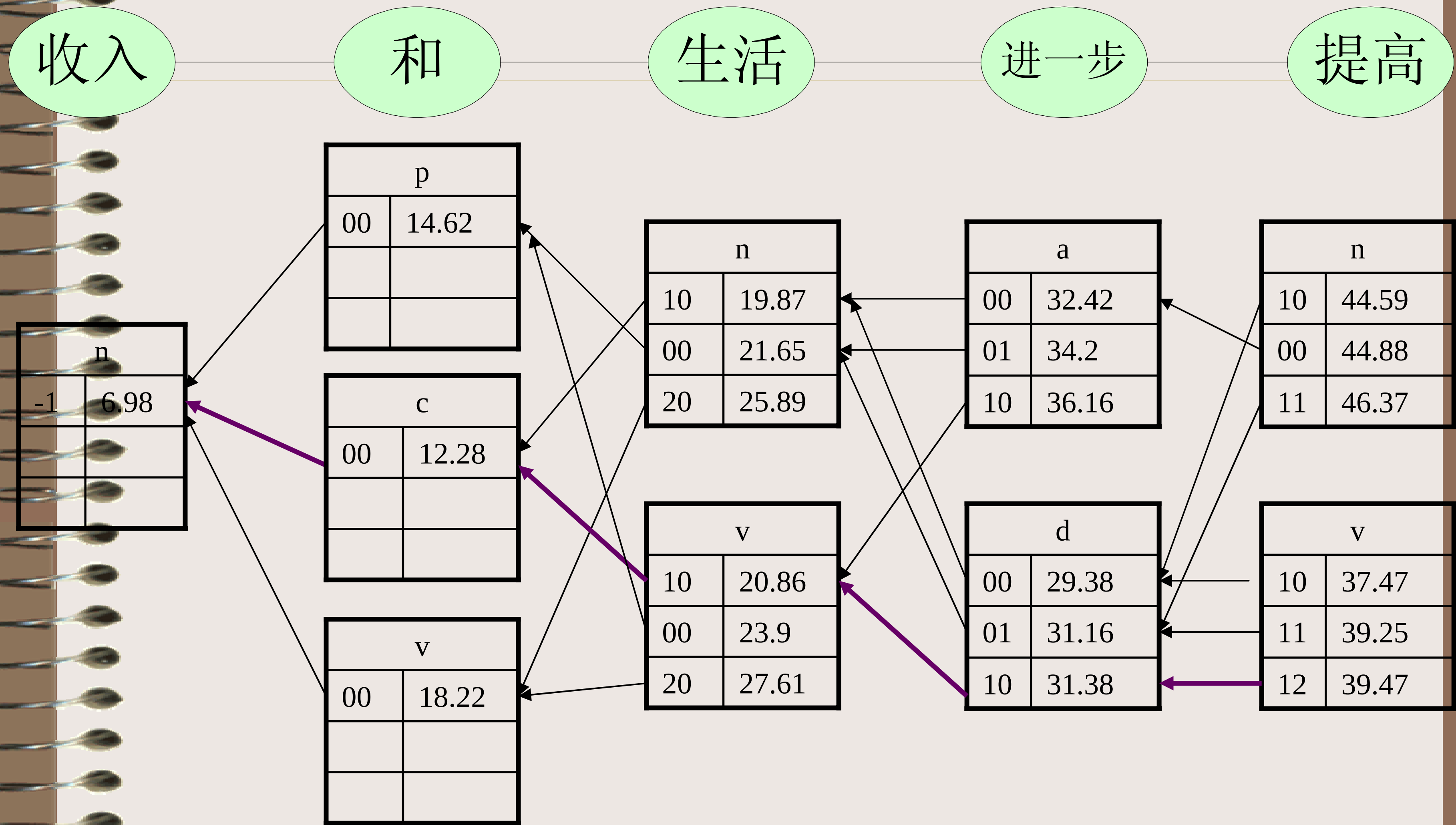


收入
和
生活
进一步
提高
n
-1 6.98
p
00 14.62
c
00 12.28
v
00 18.22
n
10 19.87
00 21.65
20 25.89
v
10 20.86
00 23.9
20 27.61
a
00 32.42
01 34.2
10 36.16
d
00 29.38
01 31.16
10 31.38
n
10 44.59
00 44.88
11 46.37
v
10 37.47
11 39.25
12 39.47


收入
和
生活
进一步
提高
p
00 14.62
c
00 12.28
v
00 18.22
n
-1 6.98
n
10 19.87
00 21.65
20 25.89
v
10 20.86
00 23.9
20 27.61
a
00 32.42
01 34.2
10 36.16
d
00 29.38
01 31.16
10 31.38
n
10 44.59
00 44.88
11 46.37
v
10 37.47
11 39.25
12 39.47



收入
和
生活
进一步
提高
n
-1
6.98
p
00
14.62
c
00
12.28
v
00
18.22
n
10
19.87
00
21.65
20
25.89
v
10
20.86
00
23.9
20
27.61
a
00
32.42
01
34.2
10
36.16
d
00
29.38
01
31.16
10
31.38
n
10
44.59
00
44.88
11
46.37
v
10
37.47
11
39.25
12
39.47



收入
和
生活
进一步
提高
n
-1 6.98
p
00 14.62
c
00 12.28
v
00 18.22
n
10 19.87
00 21.65
20 25.89
v
10 20.86
00 23.9
20 27.61
a
00 32.42
01 34.2
10 36.16
d
00 29.38
01 31.16
10 31.38
n
10 44.59
00 44.88
11 46.37
v
10 37.47
11 39.25
12 39.47

N-Best Search结果

1)收入/n 和/c 生活/n 进一步/d 提高/v	37.47
2)收入/n 和/p 生活/n 进一步/d 提高/v	39.25
3)收入/n 和/c 生活/v 进一步/d 提高/v	39.47

谢谢！