

Natural Language Processing

#L5

词性标注

袁彩霞

yuancx@bupt.edu.cn

人工智能学院 智能科学与技术中心

主要内容

- 词性及词性标注
- 隐马尔可夫模型
 - The Forward Algorithm
 - The Viterbi Algorithm
 - The Baum-Welch (EM Algorithm)
- 其它的序列标注任务

词性(Part-of-Speech, POS)

- 词性：词语在区别（语法功能的）词类时具有的属性
 - 名词 (noun)
 - 动词 (verb)
 - 形容词 (adjective)
 - 副词 (adverb)
 - 介词 (preposition)
 - 连词 (conjunction)
 -

形容词(a):

- 情状形容词(ad)
- 非谓形容词(an)
- 唯谓形容词(ap)
- 性质形容词(aq)
- 状态形容词(as)

区别词(b)

连词(c)

- 并立连词(cc)
- 从属连词(cs)

副词(d):

- 关联副词(dc)
- 可修饰名词性成分的副词(dn)
- 程度副词(dd)

叹词(e)

方位词(f):

- 双音节方位词(fd)
- 单音节方位词(fm)

语素字(g):

- 形容词性语素(ga)
- 名词性语素(gn)
- 动词性语素(gv)

前接成分(h):

- 数前接成分(hm)
- 名前接成分(hn)

习用语(i)

简称和略语(j):

- 形容词性简称和略语(ja)
- 名词性简称和略语(jn)
- 动词性简称和略语(jv)

后接成分(k)

- 名后接成分(kn)
- 动后接成分(kv)

数词(m):

- 基数词(mc)
- 位数词(mcw)
- 系数词(mcx)
- 序数词(mo)
- 序列词(mos)
- 数量数词(mq)
- 助数词(mu)

名词(n):

- 普通名词(ng)
- 无量名词(ngq)
- 方位名词(nl)
- 专有名词(np)
- 人名(nph)
- 团体机构名(npi)
- 地名(npp)
- 处所名词(ns)
- 时间名词(nt)

助词(u):

- 动态助词(ua)
- 比况助词(uc)
- 语气助词(um)
- 替代助词(ur)
- 结构助词(us)

拟声词(o)

介词(p)

量词(q)

- 名量词(qn)
- 复合量词(qnc)
- 不定量词(qni)
- 度量词(qnm)
- 个体量词(qns)
- 时量词(qt)
- 动量词(qv)

代词(r)

动词(v):

- 趋向动词(vd):
- 不及物动词(vi)
- 系动词(vl)
- 及物动词(vt)
- 兼语动词(vtc)
- 双宾动词(vtd)
- 形式动词(vtf)
- 体宾动词(vtn)
- 小句宾动词(vts)
- 助动词(vu)

其他(w)

- 阿拉伯数字串(wd)
- 其他符号(wo)
- 中文标点符号(wp)
- 未知词(wu)

非语素字(x)

- 语气词(y)
- 状态词(z)

为什么要研究词性

- 词性是词的一个重要属性
- 有助于其它的NLP任务 (more than you'd think)
 - Text-to-speech: record, lead
 - Lemmatization: saw[v] → see, saw[n] → saw
 - NP-chunk detection: `grep {JJ | NN}* {NN | NNS}`
 - Word sense disambiguation: can (noun or modal verb)

词性标注中的歧义

- 往往存在一个词具有不止一个词性的情况
 - 词的兼类
 - 词性歧义
- “Like” 可以是动词V或介词P
 - I **like/V** candy.
 - Time flies **like/P** an arrow.
- “比较” 可以是动词V或副词ADV或动名词VN
 - 与国内相 **比较/V** , 这里的实验条件好
 - 结构 **比较/ADV** 特殊
 - 通过 **比较/VN** , 可以得出结论

词性标注中的歧义

- 以英文中的Brown corpus为例：
 - 11.5%的词（word types）具有词性歧义
 - 40%的词符（tokens）具有词性歧义
- 以英文中的Penn treebank为例：
 - 不同的人类专家在进行词性标注过程中存在着3.5%的歧义
- 汉语缺乏语法形态变化，词的应用非常灵活，词的兼类现象更多，也更复杂

影响词性标注的因素

- 词本身：

- 一些词只具有某一个特定的词性，例如：天安门
- 一些词的词性具有歧义，例如：把，比较
- 若一个词的某个词性比其它词性出现更多，则出现概率对判断该词的词性会有影响
 - 一个基线：为一些兼类词挑选最常用的词性，可以达到大约90%的准确率

- 上下文：

- 两个定冠词连续出现的情况很少
- 两个基础动词连续出现的情况很少
- 定冠词后跟形容词或名词的可能性较大
-

词性标注

- 词性标注的方法：
 - **Rule-Based (基于规则的方法)**: 人类专家根据词法及语言学知识编制的规则
 - IF(...) THEN(...)
 - **Learning-Based (基于学习的方法)**: 从专家标注的语料库中学习用于自动标注的模型
 - 统计模型: 隐马尔可夫模型 (HMM), 条件随机域模型 (CRF), 神经网络模型 (NN)
 - 规则学习: 基于转换的学习 (TBL)
- 重点介绍: 隐马尔可夫模型

词性标注

- 学习过程:

- 给定训练数据: $(O_i, Q_i), i=1, \dots, N$

- 对于词性标注任务: N 为训练数据中的句子总数, O_i 为训练数据中的第 i 个句子 (词序列), Q_i 为 O_i 对应的词性序列

- 目标: 根据训练数据, 得到一个函数 $f(O)$, 完成从输入 O 到其标记 $f(O)$ 的映射

- 对于词性标注任务: 完成从输入的词序列 O 到词性序列 $f(O)$ 的映射

词性标注

- 标注过程:

- 输入：句子（词序列）

- Profits soared at Boeing Co., easily topping forecasts on Wall Street, as their CEO Alan Mulally announced first quarter results.
 - （对于中文来说，输入为切好词的句子）

- 输出：句子中每个词的词性（词性序列）

- Profits/**N** soared/**V** at/**P** Boeing/**N** Co./**N** ,/, easily/**ADV** topping/**V** forecasts/**N** on/**P** Wall/**N** Street/**N** ,/, as/**P** their/**POSS** CEO/**N** Alan/**N** Mulally/**N** announced/**V** first/**ADJ** quarter/**N** results/**N** ./.

词性标注

- 目标：为词序列O找到最优的词性标记序列Q

$$\arg \max_Q P(Q|O)$$

– 即：求解使得概率 $P(Q|O)$ 最大的Q

- 根据Bayes法则：

$$P(Q|O) = \frac{P(O|Q)P(Q)}{P(O)}$$

- 由于仅关心 $\arg \max_Q$ ，可以将上式中的 $P(O)$ 忽略：

$$\arg \max_Q P(Q|O) = \arg \max_Q P(O|Q) P(Q)$$

- 如何求解？

模型分解

$$\arg \max_Q P(Q|O) = \arg \max_Q P(O|Q) P(Q)$$

- $P(O|Q)$ 可以被分解为：

$$P(O|Q) = P(o_1, \dots, o_n | q_1, \dots, q_n) \approx \prod_i P(o_i | q_i)$$

- $P(Q)$ 为词性上的语言模型，可以通过n-gram model进行计算，例如bigram：

$$P(Q) = P(q_1, \dots, q_n) \approx P(q_1) P(q_2 | q_1) P(q_3 | q_2) \dots P(q_n | q_{n-1})$$

- 至此，我们已经得到了隐马尔可夫模型的雏形

模型分解

$$\arg \max_Q P(Q|O) = \arg \max_Q P(O|Q) P(Q)$$

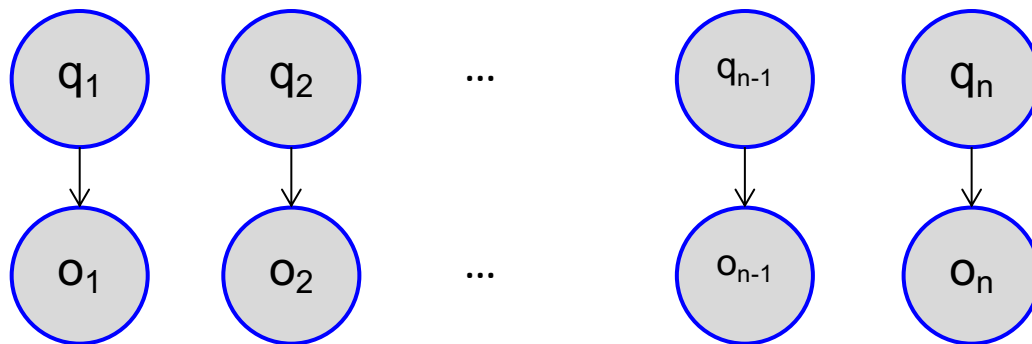
- 另外一个角度：图模型

$$P(Q) = P(q_1, \dots, q_n) \approx P(q_1) P(q_2|q_1) P(q_3|q_2) \dots P(q_n|q_{n-1})$$



词性间的马尔可夫链

$$P(O|Q) = P(o_1, \dots, o_n | q_1, \dots, q_n) \approx \prod_i P(o_i | q_i)$$



词性 \rightarrow 词的生成关系

马尔可夫链 (Markov chain)

- 一组状态：

- $Q = q_1, q_2 \dots q_N$; t 时刻的状态为 q_t

- 一组转移概率：

- $A = a_{11}a_{12} \dots a_{N1} \dots a_{NN}$

- 每个元素 a_{ij} 表示从状态 i 到状态 j 的转移概率

- 由 a_{ij} 构成的状态转移矩阵 A

$$a_{ij} = P(q_t = j \mid q_{t-1} = i) \quad 1 \leq i, j \leq N$$

$$\sum_{j=1}^N a_{ij} = 1; \quad 1 \leq i \leq N$$

马尔可夫链 (Markov chain)

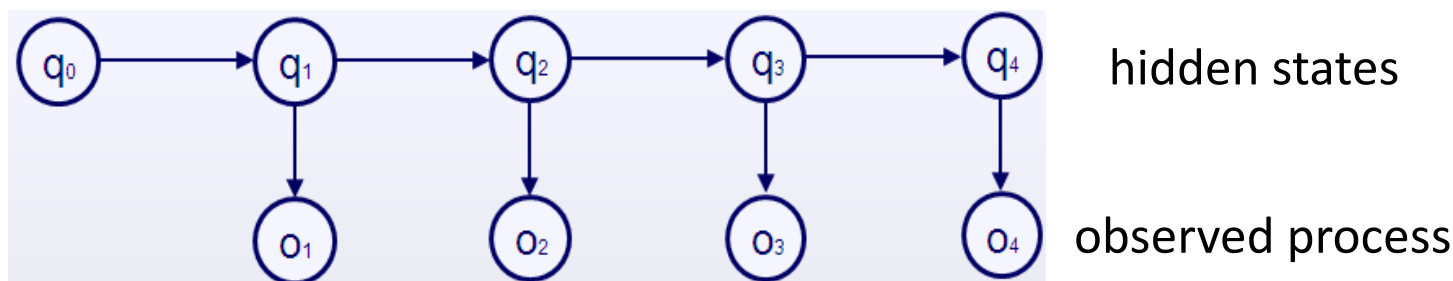
- 起始状态：初始状态的概率向量 $\pi=(\pi_1, \dots, \pi_N)$
 - 表示各状态作为初始状态的概率
- 约束：

$$\pi_i = P(q_1 = i) \quad 1 \leq i \leq N$$

$$\sum_{j=1}^N \pi_j = 1$$

隐马尔可夫模型

- 图表示:



- 两个假设:

- Markov假设:

$$P(q_i | q_1 \dots q_{i-1}) = P(q_i | q_{i-1})$$

- 输出独立性假设:

$$P(o_t | O_1^{t-1}, q_1^t) = P(o_t | q_t)$$

隐马尔可夫模型

$$Q=q_1q_2\ldots q_N$$

N个**状态**

$$A=a_{11}a_{12}\ldots a_{1N}\ldots a_{NN}$$

状态转移矩阵A，每个元素 a_{ij} 表示从状态 i 到状态 j 的转移概率，满足对于任意 i ， $\sum_{j=1,\ldots,N} a_{ij}=1$

$$O=o_1o_2\ldots o_T$$

T个**观测**，每个观测都来自于词典 $V=v_1, v_2, \ldots, v_{|V|}$

$$B=b_i(o_t)$$

发射概率矩阵B，每个元素 $b_i(o_t)$ 表示观测的似然，即由状态 i 产生观测 o_t 的概率

$$q_0, q_F$$

起始状态和终止状态的概率，不和任何观测关联，从起始状态的对应的转移概率 $a_{01}a_{02}\ldots a_{0N}$ 出发，到终止状态对应的概率 $a_{1F}a_{2F}\ldots a_{NF}$ 结束

一个实例

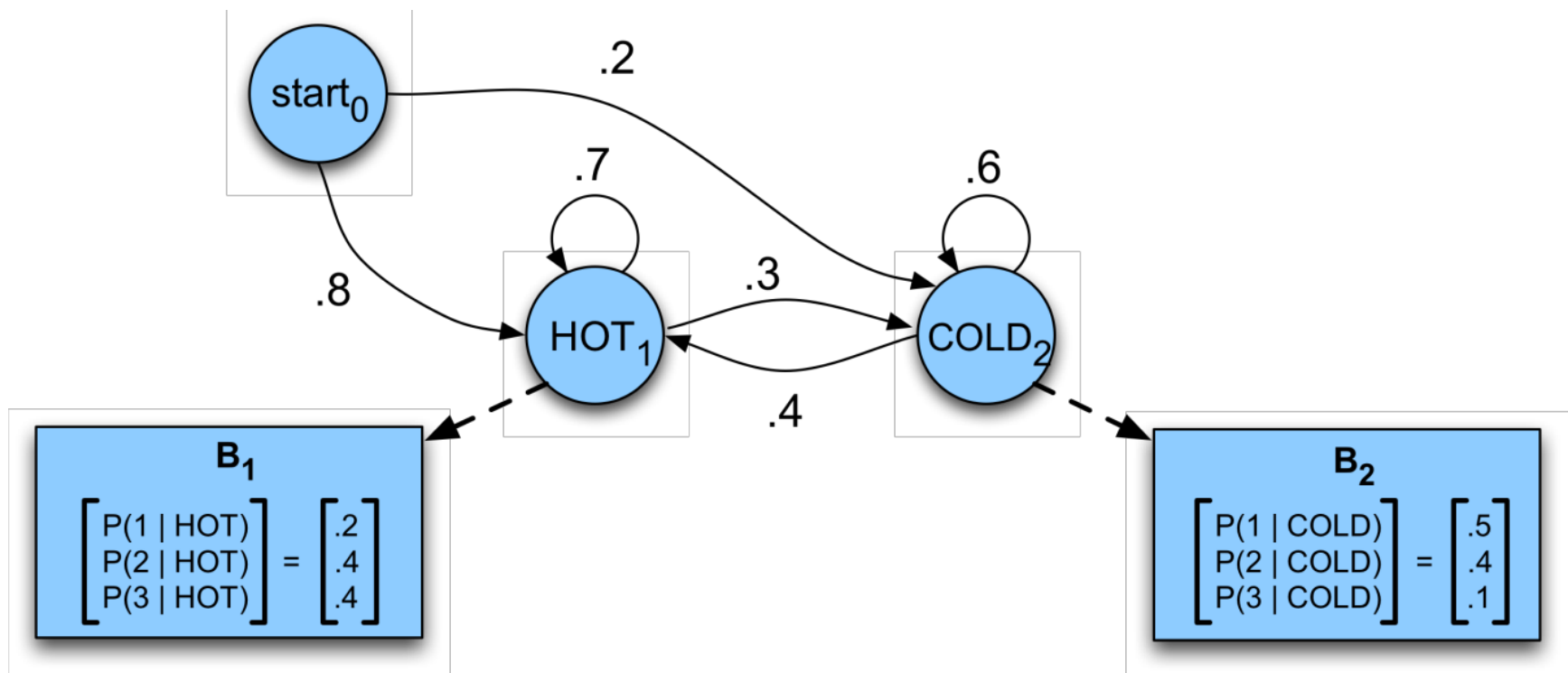
- 考察一个比词性标注任务相似但略为简单的问题
- 问题设置：
 - 假设你是一个在2217年研究气候变暖的气象学家
 - 却找不到北京2017年夏天的天气数据记录
 - 但幸运的是Mr. Lee通过日记记录了这个夏天每天吃了多少根冰淇淋
 - 你的工作：推算出这个夏天有多热
- 给定：
 - Ice Cream序列（观测）：1, 2, 3, 2, 2, 2, 3... (词序列)
- 求解：
 - Weather 序列（状态）：Hot, Cold, Hot, Hot, Hot, Cold... (词性序列)

一个实例

- 假设:

每天可以吃的冰淇淋数目可以为1、2、3 (观测集合)

天气状态为hot (状态1) 或cold (状态2) (状态集合)



HMM能够建模的三个基本问题

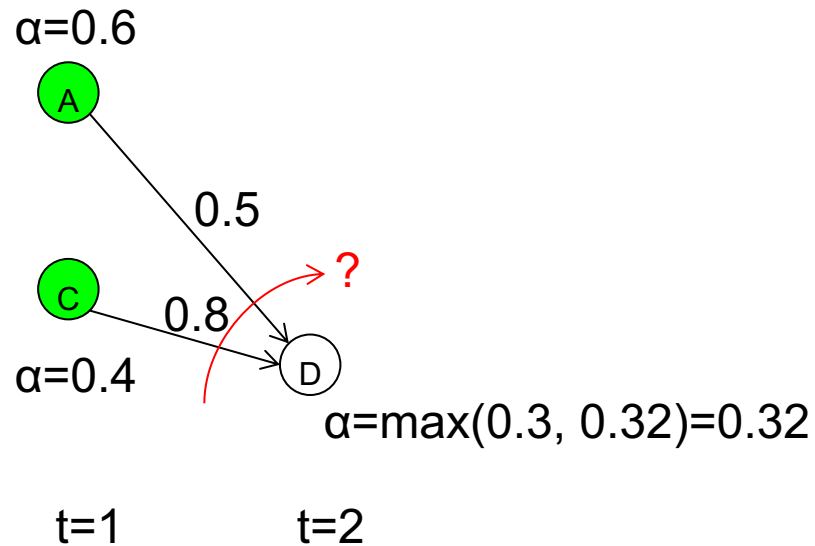
- Problem 1 (估算问题):
 - 给定观测序列 $O=(o_1o_2...o_T)$, 及HMM模型参数 $\lambda=(A,B)$
 - 如何计算 $P(O|\Phi)$, 即计算产生某个观测序列的概率 (观测的似然)
- Problem 2 (解码问题):
 - 给定观测序列 $O=(o_1o_2...o_T)$, 及HMM模型参数 $\lambda=(A,B)$
 - 如何计算最优的状态序列 $Q=(q_1q_2...q_T)$ (i.e., 能最好解释观测 O 的状态序列)
- Problem 3 (参数学习):
 - 如何学习模型参数 $\lambda=(A,B)$ 使 $P(O|\lambda)$ 最大化

词性标注问题：解码

- **问题：** 给定观测序列 $O=(o_1o_2...o_T)$ ，以及 HMM 模型参数 $\lambda=(A,B)$ ，计算其对应的最可能的状态序列 $Q=(q_1q_2...q_T)$ (i.e., 能最好的解释观测)
- 例：已知 Ice cream 观测序列 3-1-3，以及 HMM
- 解码的任务：
 - 找到观测序列 3-1-3 背后最可能的天气状态序列 (H-C-H? H-H-C?)

- 一种可能：
 - 对于一个观测序列 O
 - E.g., 3-1-3
 - 计算 $P(Q|O)$
 - E.g., $P(HHH|313)$, $P(HHC|313)$, ...
 - 从中选取一个概率最大值对应的状态序列
- Why not?
 - N^T
- 一个更高效的方法：the Viterbi algorithm
 - 线性规划算法
 - 使用格栅记录计算过程

Viterbi 算法：基本思路



- 直觉：
 - C 应该是到达D的最可能状态，因为
 - $p(A-D)=0.6 \times 0.5=0.3$
 - $p(C-D)=0.4 \times 0.8=0.32$
- 尽管没有进行全部的搜索，D已经找到它最可能的前一个状态

Viterbi 算法：基本思路

- $Q_{\text{best}} = \arg \max_Q P(Q|O)$
 $= \arg \max_Q P(Q,O)/P(O)$
 $= \arg \max_Q P(Q,O)$

→ 求解观测序列及最佳状态序列的联合概率

$$= \arg \max_Q \prod_{i=1 \dots t} P(o_i | q_i, q_{i-1}) P(q_i | q_{i-1})$$

- 定义 $v_t(j)$: t 时刻到达状态 j 的最可能路径对应的概率

$$v_t(j) = \max_{q_0, q_1, \dots, q_{t-1}} P(q_0, q_1, \dots, q_{t-1}, o_1, o_2, \dots, o_t, q_t = j | \lambda)$$

Viterbi 算法：基本思路

- 从左向右处理观测序列
- 将 $v_t(j)$ 填充到对应的格子中
- 递归：

$$v_t(j) = \max_{i=1}^N v_{t-1}(i) a_{ij} b_j(o_t)$$

a_{ij} : 从状态 i 到状态 j 的转移概率

$b_i(o_t)$: 由状态 i 产生观测 o_t 的发射概率

$v_t(j)$: t 时刻时到达状态 j 的viterbi路径的概率

Viterbi 算法：前向递归过程

- 初始化： $v_1(j) = a_{0j}b_j(o_1), 1 \leq j \leq N$

$$bt_1(j) = 0$$

- 递归：

$$v_t(j) = \max_{i=1}^N v_{t-1}(i)a_{ij}b_j(o_t), 1 \leq j \leq N, 1 < t \leq T$$

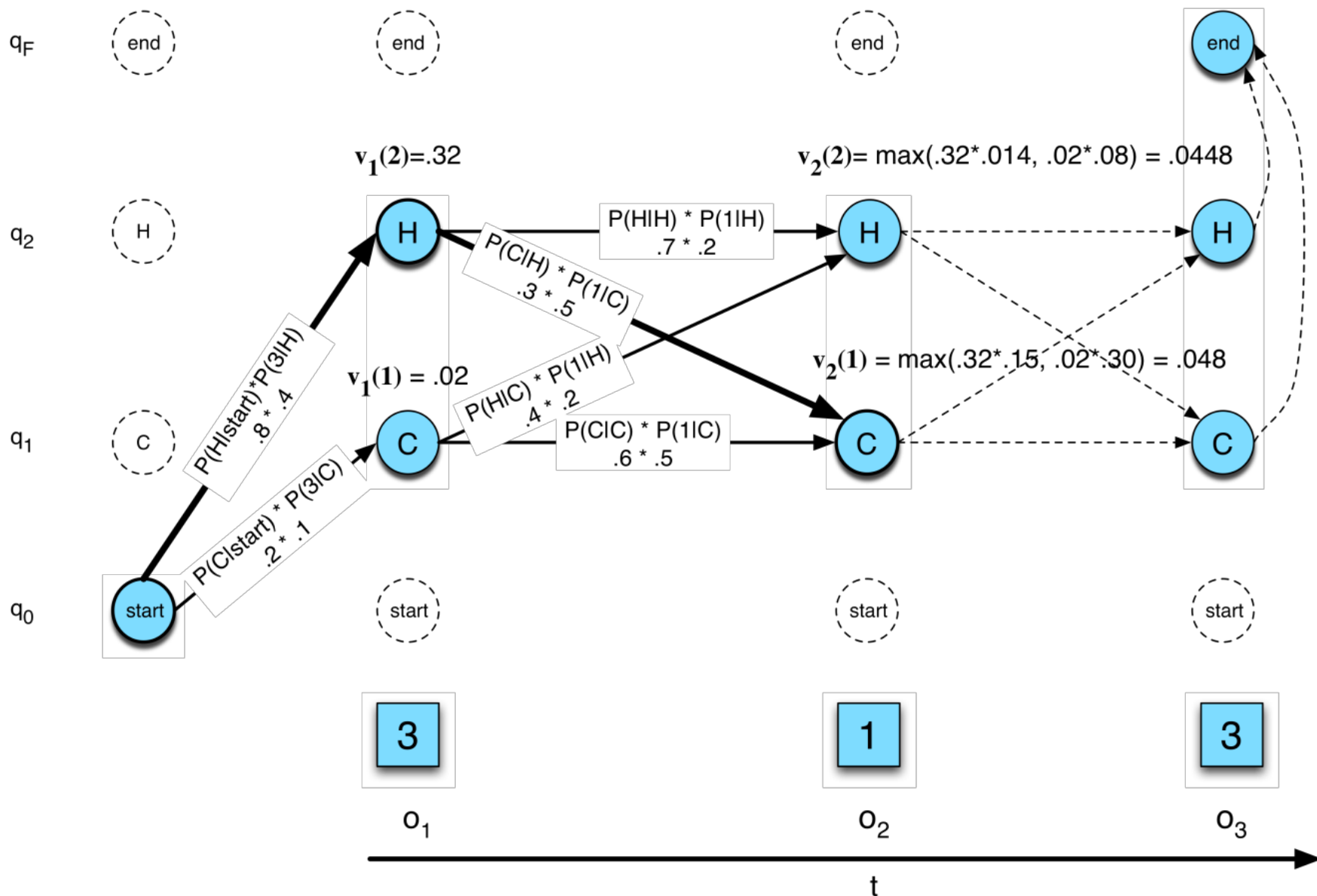
$$bt_t(j) = \arg \max_{i=1}^N v_{t-1}(i)a_{ij}b_j(o_t), 1 \leq j \leq N, 1 < t \leq T$$

- 终止：

最优路径对应的概率： $P^* = v_T(q_F) = \max_{i=1}^N v_T(i)a_{iF}$

回退的起始状态： $q_T^* = bt_T(q_F) = \arg \max_{i=1}^N v_T(i)a_{iF}$

Viterbi 算法：格栅



Viterbi 算法

function VITERBI(*observations* of len T , *state-graph* of len N) **returns** *best-path*

create a path probability matrix $viterbi[N+2, T]$

for each state s **from** 1 **to** N **do** ; initialization step

$viterbi[s, 1] \leftarrow a_{0,s} * b_s(o_1)$

$backpointer[s, 1] \leftarrow 0$

for each time step t **from** 2 **to** T **do** ; recursion step

for each state s **from** 1 **to** N **do**

$viterbi[s, t] \leftarrow \max_{s'=1}^N viterbi[s', t-1] * a_{s',s} * b_s(o_t)$

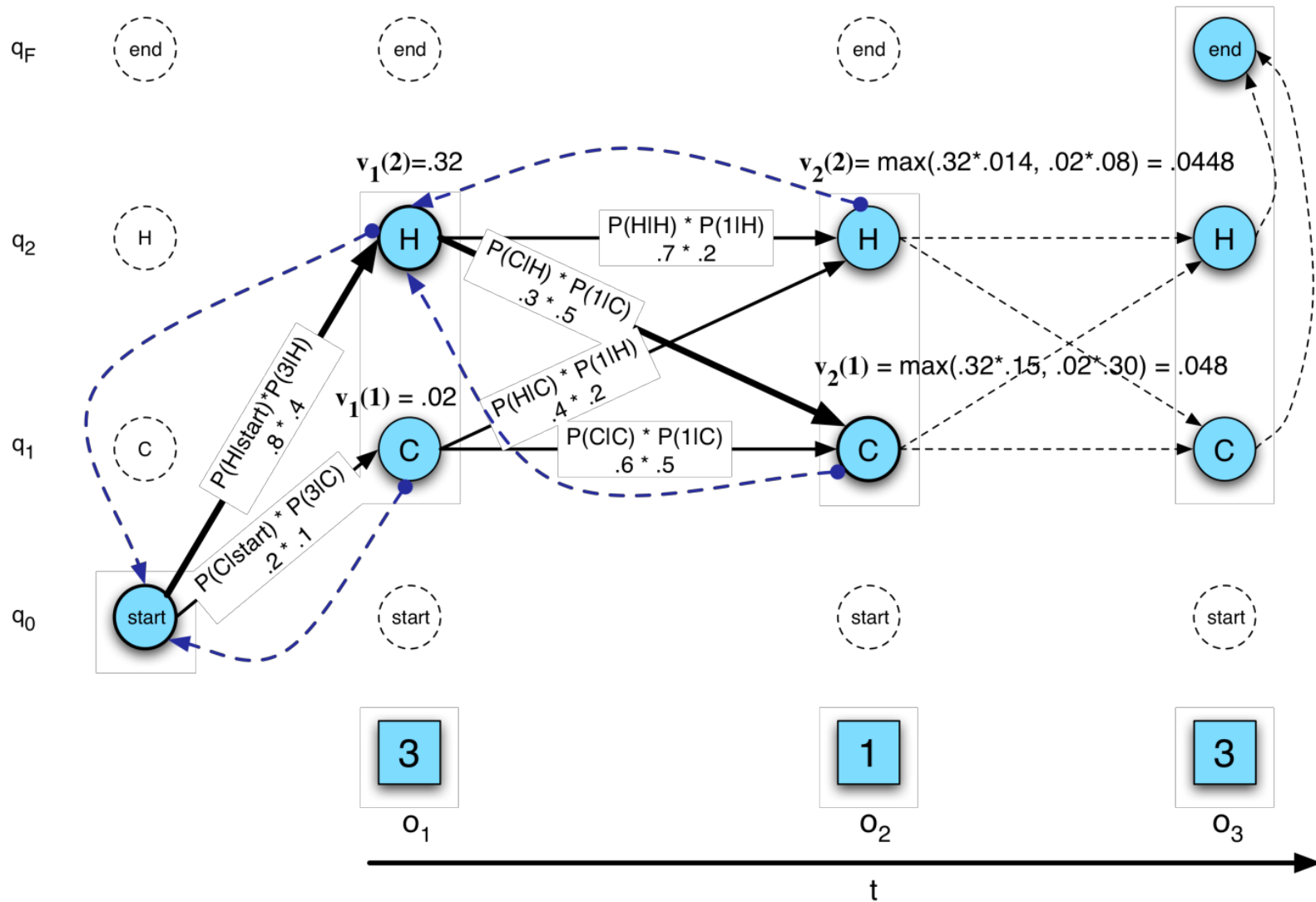
$backpointer[s, t] \leftarrow \operatorname{argmax}_{s'=1}^N viterbi[s', t-1] * a_{s',s}$

$viterbi[q_F, T] \leftarrow \max_{s=1}^N viterbi[s, T] * a_{s,q_F}$; termination step

$backpointer[q_F, T] \leftarrow \operatorname{argmax}_{s=1}^N viterbi[s, T] * a_{s,q_F}$; termination step

return the backtrace path by following backpointers to states back in time from $backpointer[q_F, T]$

Viterbi 算法：回退



模型训练：参数学习

- **问题：**如何找到能最好解释观测数据的模型参数 $\lambda=(A, B, \Pi)$

$$\arg \max_{\lambda} P(O_{training} | \lambda)$$

- 进一步：如何从数据中自动估计模型参数？
- 一个简单的方法：极大似然估计法

参数学习

- 有监督方法：基于已知“正确答案”的数据（序列对）进行估计
 - 从数据中统计状态之间的转移，及状态到观测的发射
 - 记 $n=1, \dots, N$ 为有标记样本数， $t=1, \dots, T$ 为每个样本的序列长度， $q_{n,t}^i$ 为第 n 个序列在 t 时刻的状态， $o_{n,t}^k$ 为第 n 个序列在 t 时刻的观测

$$a_{ij}^{MLE} = \frac{\#(i \rightarrow j)}{\#(i \rightarrow *)} = \frac{\sum_{n=1}^N \sum_{t=2}^T q_{n,t-1}^i q_{n,t}^j}{\sum_{n=1}^N \sum_{t=2}^T q_{n,t-1}^i}$$
$$b_{jk}^{MLE} = \frac{\#(j \rightarrow k)}{\#(j \rightarrow *)} = \frac{\sum_{n=1}^N \sum_{t=1}^T q_{n,t}^j o_{n,t}^k}{\sum_{n=1}^N \sum_{t=1}^T q_{n,t}^j}$$

参数学习

- 无监督方法：基于无“正确答案”的数据（不完全数据）进行估计
- 采用MLE，意味着需要找到使下列概率最大的参数 λ ：

$$\arg \max_{\lambda} P(O_{training} | \lambda)$$

- 很难找到上式的解析解！
- 但可以通过类似于爬山算法的局部最大化方法求解 → Baum-Welch算法 or Forward-Backward算法

前向-后向算法

- 基本思想:

- 不知道模型参数 λ ，但可以通过某个参数计算观测的似然概率 $P(O|\lambda')$
 - 比如可以随机选取一个 λ'
- 通过计算，可以看到哪些状态转移和观测发射最可能被使用到
- 增大它们的概率，可以得到一个新的模型 λ'' ，新的模型赋予观测序列一个更高的似然概率

前向-后向算法

- 定义前向变量 $\alpha_t(j)$

$$\alpha_t(j) = P(o_1, o_2, \dots, o_t, q_t = j \mid \lambda)$$

- 表示已知前t个时刻的观测序列，同时，在t时刻位于状态j的概率
- 该值被记录在格栅中的 (s_j, t) 位置

- 初始化：

$$\alpha_1(j) = a_{0j} b_j(o_1), 1 \leq j \leq N$$

- 递归：

$$\alpha_t(j) = \sum_{i=1}^N \alpha_{t-1}(i) a_{ij} b_j(o_t), 1 \leq j \leq N, 1 < t \leq T$$

(θ 状态和 F 状态不对应任何观测)

前向-后向算法

- 定义后向变量 $\beta_t(i)$:

$$\beta_t(i) = P(o_t, \dots, o_T \mid q_t = i, \lambda)$$

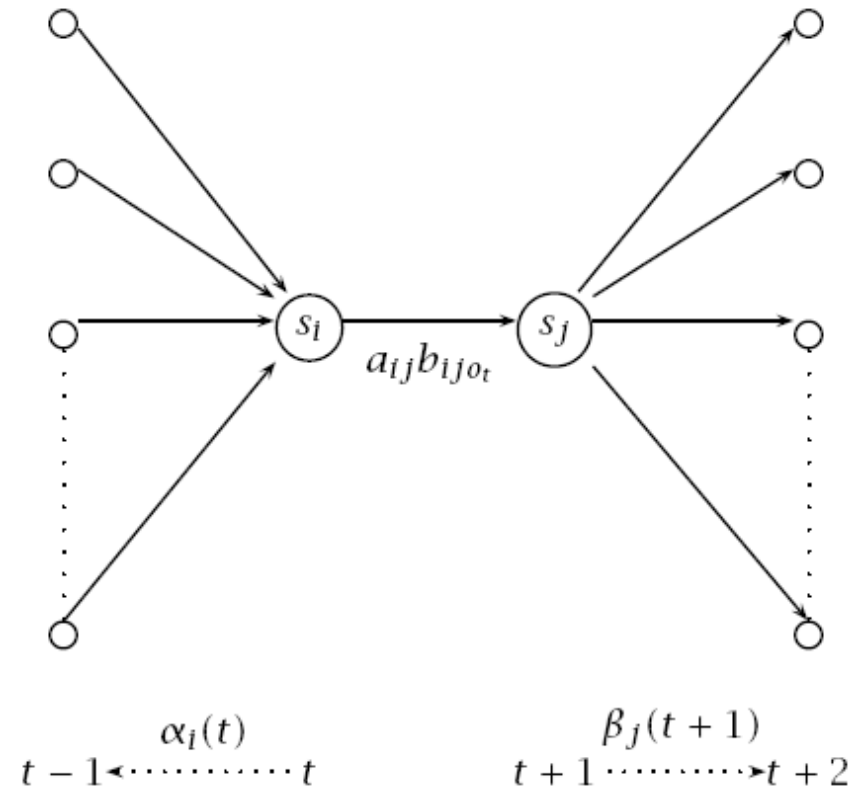
- 即：在 t 时刻位于状态 i 时，得到 t 时刻后的观测的概率
- 初始化：

$$\beta_{T+1}(i) = 1, 1 \leq i \leq N$$

- 递归：

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(o_t) \beta_{t+1}(j), \quad 1 \leq j \leq N, \quad 1 \leq t < T$$

前向-后向算法



$$\alpha_i(t) = P(o_1, o_2, \dots, o_{t-1}, q_t = i \mid \lambda)$$

$$\alpha_j(t+1) = \sum_{i=1}^N \alpha_i(t) a_{ij} b_j(o_t)$$

$$\beta_j(t+1) = P(o_{t+1} \dots o_T \mid q_{t+1} = j, \lambda)$$

$$\beta_i(t) = \sum_{j=1}^N a_{ij} b_j(o_t) \beta_j(t+1)$$

前向-后向算法

- 前向算法和后向算法结合:

$$\begin{aligned}P(O, q_t = i \mid \lambda) &= P(o_1, \dots, o_T, q_t = i \mid \lambda) \\&= P(o_1, \dots, o_{t-1}, q_t = i, o_t, \dots, o_T \mid \lambda) \\&= P(o_1, \dots, o_{t-1}, q_t = i \mid \lambda) \times P(o_t, \dots, o_T \mid o_1, \dots, o_{t-1}, q_t = i, \lambda) \\&= P(o_1, \dots, o_{t-1}, q_t = i \mid \lambda) \times P(o_t, \dots, o_T \mid q_t = i, \lambda) \\&= \alpha_t(i) \beta_t(i)\end{aligned}$$

- 因此:

$$P(O \mid \lambda) = \sum_{i=1}^N \alpha_t(i) \beta_t(i), 1 \leq t \leq T + 1$$

前向-后向算法

定义 $p_t(i, j)$:

已知观测序列 O , t 时刻从状态 i 转移到状态 j 的概率

$$\begin{aligned} p_t(i, j) &= P(q_t = i, q_{t+1} = j | O, \lambda) \\ &= \frac{P(q_t = i, q_{t+1} = j, O | \lambda)}{P(O | \lambda)} \\ &= \frac{\alpha_i(t) a_{ij} b_j(o_t) \beta_j(t+1)}{\sum_{m=1}^N \alpha_m(t) \beta_m(t)} \\ &= \frac{\alpha_i(t) a_{ij} b_j(o_t) \beta_j(t+1)}{\sum_{m=1}^N \sum_{n=1}^N \alpha_m(t) a_{mn} b_n(o_t) \beta_n(t+1)} \end{aligned}$$

定义 $\gamma_i(t)$:

已知观测序列 O , t 时刻位于状态 i 的概率

$$\gamma_i(t) = \sum_{j=1}^N p_t(i, j) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{i=1}^N \alpha_t(i) \beta_t(i)}$$

前向-后向算法

- 已知观测 O ，对于时间叠加，可以得到：
 - 由状态 i 转移到状态 j 的次数的期望： $\sum_{t=1, \dots, T} P_t(i, j)$
 - 由状态 i 进行状态转移的次数的期望： $\sum_{t=1, \dots, T} \gamma_i(t)$

前向-后向算法

- 模型参数更新:

$$\begin{aligned}\hat{\pi}_i &= \text{expected frequency in state } i \text{ at time } t = 1 \\ &= \gamma_i(1)\end{aligned}$$

$$\begin{aligned}\hat{a}_{ij} &= \frac{\text{expected number of transitions from state } i \text{ to } j}{\text{expected number of transitions from state } i} \\ &= \frac{\sum_{t=1}^T p_t(i, j)}{\sum_{t=1}^T \gamma_i(t)}\end{aligned}$$

$$\begin{aligned}\hat{b}_{ijk} &= \frac{\text{expected number of transitions from } i \text{ to } j \text{ with } k \text{ observed}}{\text{expected number of transitions from } i \text{ to } j} \\ &= \frac{\sum_{\{t: o_t=k, 1 \leq t \leq T\}} p_t(i, j)}{\sum_{t=1}^T p_t(i, j)}\end{aligned}$$

前向-后向算法

- **初始化**：给模型一个初始值 $\lambda = (A, B, \pi)$
- **迭代**：
 - 采用当前的模型通过O估计参数期望（期望）
 - 更新模型参数：极大似然估计法（最大化）
- **收敛**：重复以上过程，直至模型收敛到一个最优值 $\lambda^* = (A^*, B^*, \pi^*)$
 - Baum证明了模型的收敛性，即：

$$P(O|\lambda^*) \geq P(O|\lambda)$$

□

前向-后向算法

- 期望最大化算法(**Expectation Maximization (EM)**)的一个特例
- 通过迭代可以改进模型参数
- 但这种参数迭代计算的过程不保证能找到参数的最优解

- Part-of-speech tagging

- 8 个常用的英文词性类别

- N : noun , *chair, bandwidth, pacing*
 - V : verb , *study, debate, munch*
 - ADJ : adj , *purple, tall, ridiculous*
 - ADV : adverb , *unfortunately, slowly,*
 - P : preposition , *of, by, to*
 - PRO : pronoun , *I, me, mine*
 - DT : determiner , *the, a, that, those*

HMM应用于POS Tagging

WORD	tag
------	-----

the	DT
------------	-----------

koala	N
--------------	----------

put	V
------------	----------

the	DT
------------	-----------

keys	N
-------------	----------

on	P
-----------	----------

the	DT
------------	-----------

table	N
--------------	----------

HMM应用于POS Tagging

- 给定句子（观测词序列）
 - *Secretariat is expected to race tomorrow*
 - *She promised to back the bill*
- 如何求解最能解释当前观测的tag序列？
- 从概率模型的角度：
 - 考察所有可能的tag序列
 - 从中选取“给定词序列，最可能的tag序列”，即给定观测的词 $w_1...w_n$ 序列，找到其对应的最可能的tag序列 $t_1...t_n$ ，使得 $P(t_1...t_n | w_1...w_n)$ 最大：

$$\hat{t}_1^n = \arg \max P(t_1^n | w_1^n)$$

HMM应用于POS Tagging

- 由Bayes法则：

$$P(x | y) = \frac{P(y | x)P(x)}{P(y)}$$

- 得：

$$\begin{aligned}\hat{t}_1^n &= \arg \max_{t_1^n} P(t_1^n | w_1^n) \\ &= \arg \max_{t_1^n} \frac{P(w_1^n | t_1^n)P(t_1^n)}{P(w_1^n)} = \arg \max_{t_1^n} P(w_1^n | t_1^n)P(t_1^n)\end{aligned}$$

Likelihood和prior

$$\hat{t}_1^n = \arg \max_{t_1^n} \overset{\text{likelihood}}{\overleftarrow{P(w_1^n | t_1^n)}} \overset{\text{prior}}{\overleftarrow{P(t_1^n)}}$$

$$P(w_1^n | t_1^n) \approx \prod_{i=1} P(w_i | t_i)$$

$$P(t_1^n) \approx \prod_{i=1}^n P(t_i | t_{i-1})$$

$$\hat{t}_1^n = \arg \max_{t_1^n} P(t_1^n | w_1^n) \approx \arg \max_{t_1^n} \prod_{i=1}^n P(w_i | t_i) P(t_i | t_{i-1})$$

两类概率 (1)

- Tag之间的转移概率 $p(t_i | t_{i-1})$
 - 例：冠词后DT面接形容词JJ和名词NN的概率
 - *That/DT flight/NN*
 - *The/DT yellow/JJ hat/NN*
 - 极大似然估计法：通过有标记语料估计 $P(NN | DT)$

$$P(t_i | t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

$$P(NN | DT) = \frac{C(DT, NN)}{C(DT)} = \frac{56,509}{116,454} = .49$$

两类概率 (2)

- 词的似然概率： $p(w_i | t_i)$
 - 系动词VBZ 为“is”的概率
 - 极大似然估计法：通过有标记语料估计 $P(is | VBZ)$

$$P(w_i | t_i) = \frac{C(t_i, w_i)}{C(t_i)}$$

$$P(is | VBZ) = \frac{C(VBZ, is)}{C(VBZ)} = \frac{10,073}{21,627} = .47$$

一个例子：the verb "race"

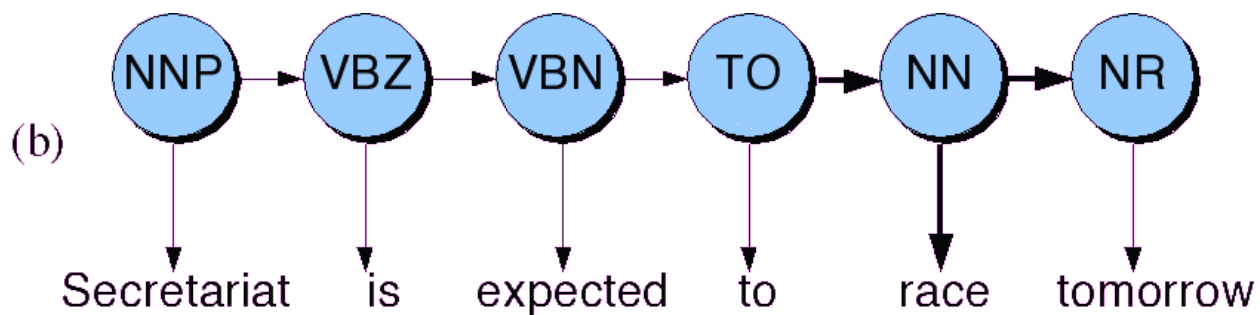
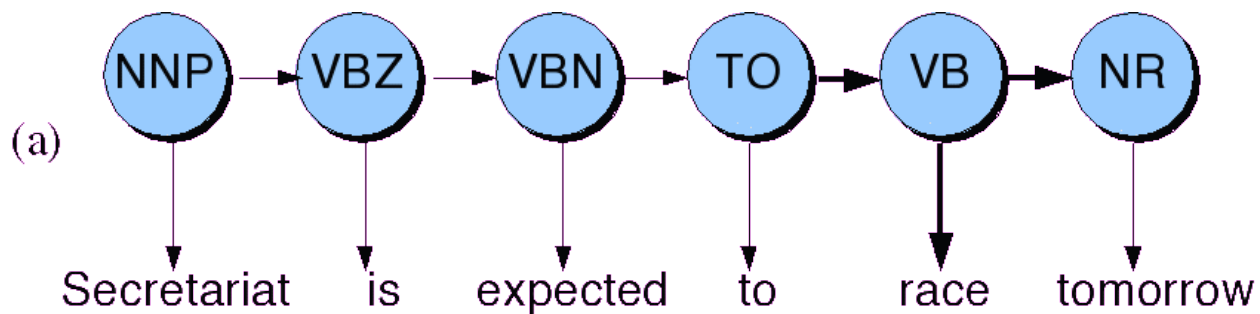
- 训练数据：

- People/**NNP** continue/**VB** to/**TO** inquire/**VB** the/**DT** reason/**NN** for/**IN** the/**DT** **race**/**NN** for/**IN** outer/**JJ** space/**NN**
- They/**NNS**
'll/**MD** never/**ADV** be/**VBZ** able/**ADJ** to/**TO** stop/**VB** using/**VBG**
/ the/**DT** word/**NN** **race**/**NN** as/**PP**
a/**DT** cultural/**ADJ** determinant/**NN**.
- Two/**QU** scientists/**NNP** **race**/**VB** for/**PP** the/**DT** prize/**NN**
- ...

- 任务：

- Secretariat will race tomorrow.
- I want to race.
- Do you know how much energy you will burn up during the relay race?
- ...

词性的歧义



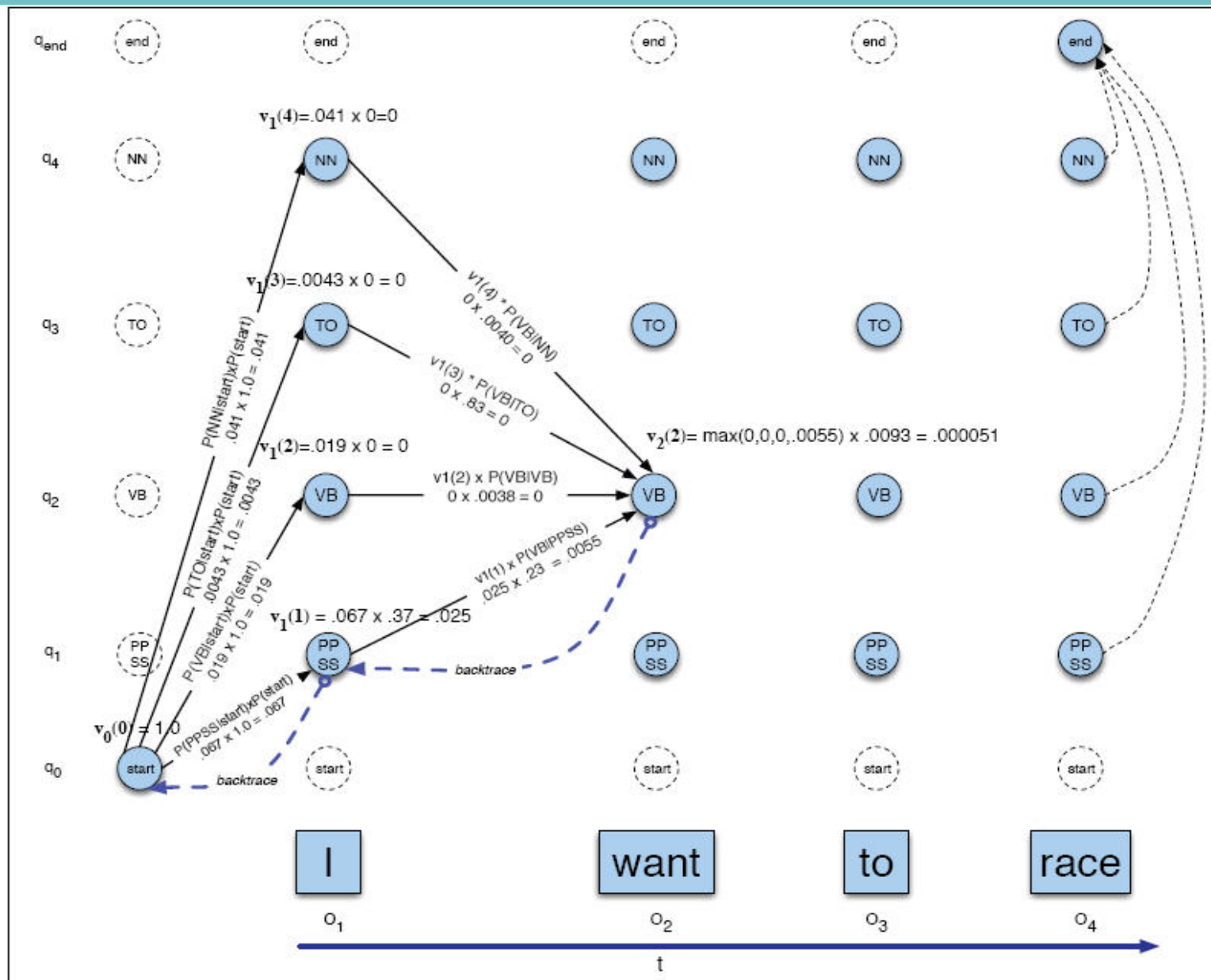
A矩阵

	VB	TO	NN	PPSS
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014

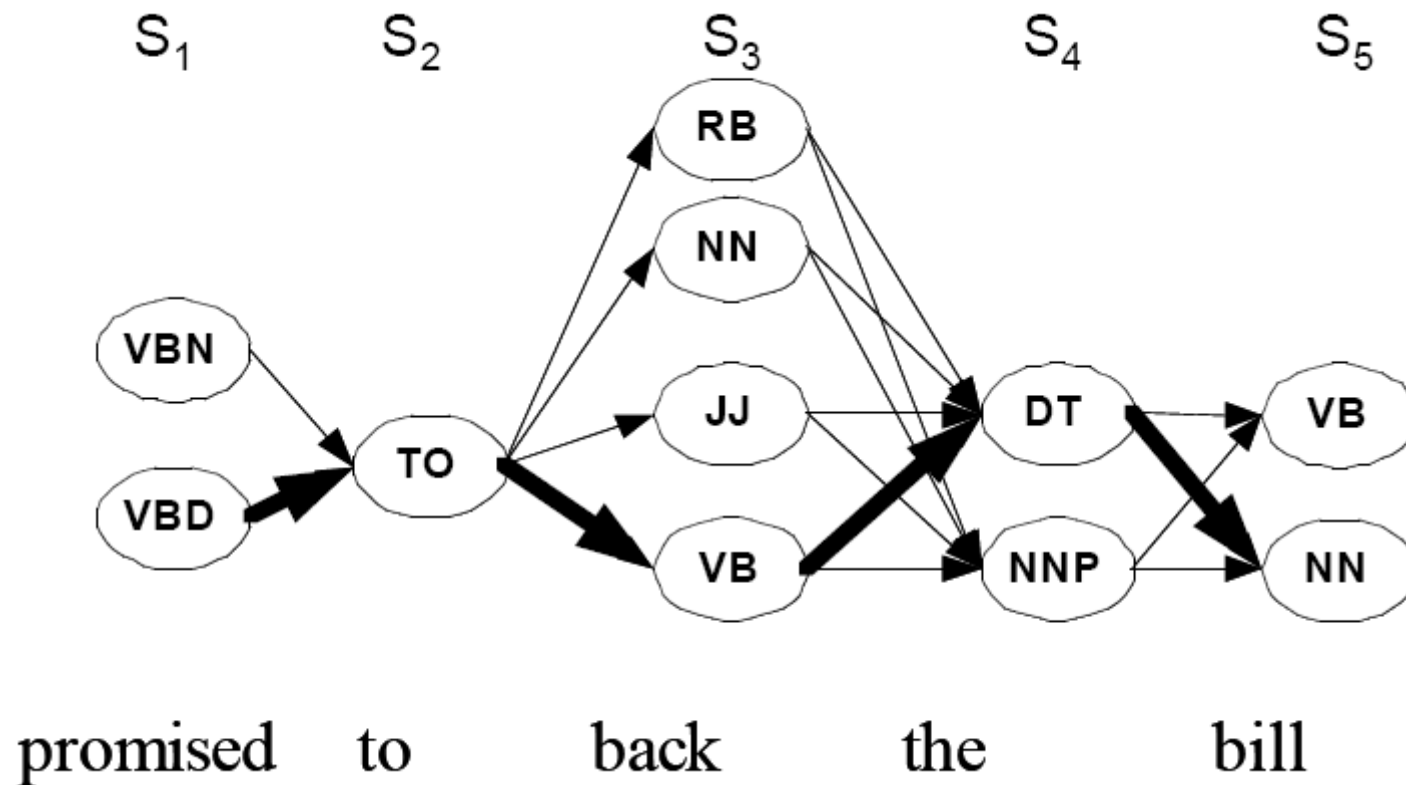
B矩阵

	I	want	to	race
VB	0	.0093	0	.00012
TO	0	0	.99	0
NN	0	.000054	0	.00057
PPSS	.37	0	0	0

Viterbi 算法：寻找最优的“路径”



Viterbi 算法：寻找最优的“路径”



其它的序列标注任务

- 分词：
 - INPUT: 富士山和涩谷风景是日本的重要标志
 - OUTPUT: 富/B 士/I 山/E 和/S 涩/B 谷/E 风/B 景/E 是/S 日/B 本/E 的/S 重/B 要/E 标/B 志/E
-

其它的序列标注任务

- 命名实体识别 (Named Entity Recognition, NER)
 - INPUT: Profits soared at Boeing Co., easily topping forecasts on Wall Street, as their CEO Alan Mulally announced first quarter results.
 - OUTPUT: Profits soared at [Company Boeing Co.], easily topping forecasts on [Location Wall Street], as their CEO [Person Alan Mulally] announced first quarter results.
- 如何将其转化为序列标注任务？
- OUTPUT: Profits/O soared/O at/O Boeing/B-C Co./I-C ,/O easily/O topping/O forecasts/O on/O Wall/B-L Street/I-L ,/O as/O their/O CEO/O Alan/B-P Mulally/I-P announced/O first/O quarter/O results/O . /O

主要内容

- 词性及词性标注
- 隐马尔可夫模型
 - The Forward Algorithm
 - The Viterbi Algorithm
 - The Baum-Welch (EM Algorithm)
- 其它的序列标注任务

Next Lecture

- 句法分析: Context-free grammar and Parsing
- Ref.: Christopher D. Manning, Ch 11, 12

- 作业提交：
 - Email至： yuansassignment@163.com
 - 邮件主题： 学号-姓名-作业题目 （e.g., 20200310-张三-语言模型）
 - 提交物包括：
 - 作业报告：任务描述、方法与模型、实验及分析、结论与思考
 - 如果是自拟题目，还需提交：实验所使用的数据