Fundamentals of Hidden Markov Model (HMM) (II) CHMM with applications to speech recognition

Ren-yuan Lyu
Dept of Computer Science & Information Engineering,
Chang Gung University,
Guei-shan, Taoyuan, Taiwan
rylyu@mail.cgu.edu.tw

Reference:

1. X. Huang, "Spoken Language Processing", Chap 8
2. L. Rabiner, "Fundamentals of Speech Recognition", Chap 6
3. HTK Book, http://htk.eng.cam.ac.uk/

Continuous HMM (CHMM)

- The (state-dependent) observation probability distribution, **B**= $\{b_i(o)\}$

By assumming $O(t)$ be a state-dependent random process,
it is enough to specify $P(O(t)=o \mid S(t)=i)$ to completely describe $O(t)$,
as long as $S(t)$ is given.

Let

$$b_i(o) = P(O(t)=o \mid S(t)=i), \qquad i \in \Omega_S = \{1,2,3,\ldots,N\},$$
$$o \in \Omega_O = R,$$

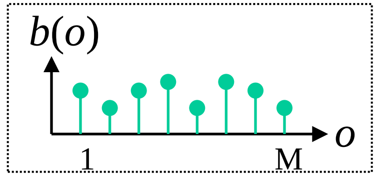
$$\underline{B} = \{b_i(o)\},$$

$$b_i(o) = \sum_{k=1}^{M} c_{ik} \cdot f\left(o;\mu_{ik},\sigma_{ik}^2\right)$$

where $f\left(o;\mu_{ik},\sigma_{ik}^2\right) = \dfrac{1}{\sqrt{2\pi\sigma_{ik}^2}} e^{-\frac{1}{2}\frac{(o-\mu_{ik})^2}{\sigma_{ik}^2}}, \ \forall i \in \Omega_S$

$$\sum_{k=1}^{M} c_{ik} = 1, \ \forall i \in \Omega_S$$

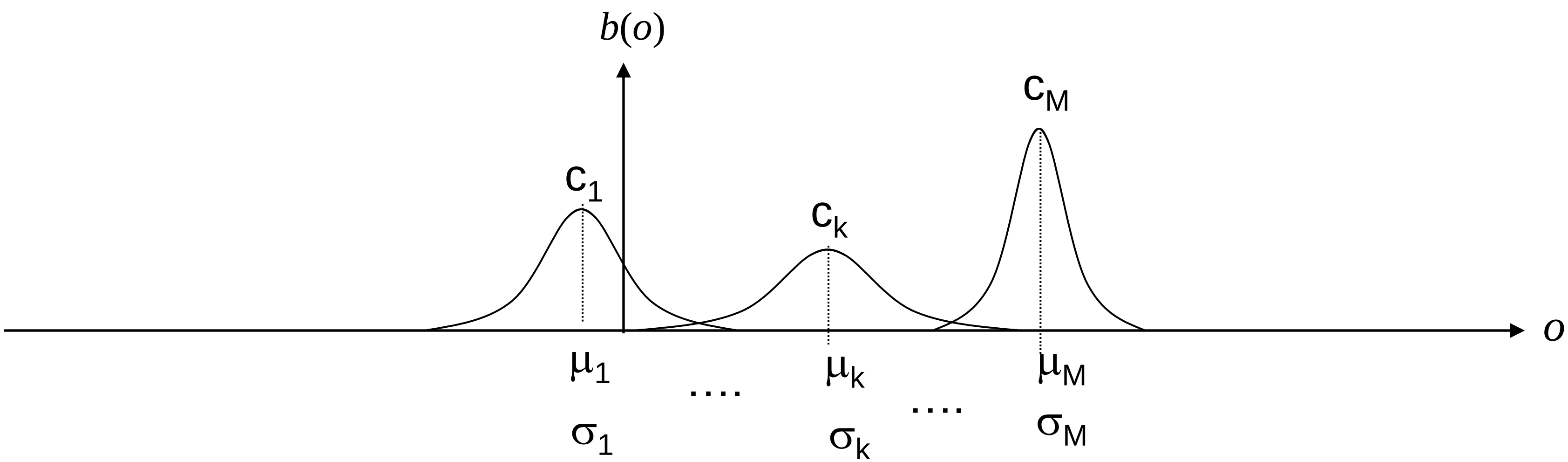
Ren-yuan Lyu
@ Chang Gung University



for each $b(o), o \in \Omega_O$

we need

$\{b_k \mid k \in [1,2,..,M]\}$



$$b(o) = \sum_{k=1}^{M} c_k \cdot f\left(o; \mu_k, \sigma_k^2\right)$$

$$\text{where } f\left(o; \mu_k, \sigma_k^2\right) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{1}{2}\frac{(o-\mu_k)^2}{\sigma_k^2}}$$

for each $b(o), o \in \Omega_O$

we need

$\{c_k, \mu_k, \sigma_k^2 \mid k \in [1,2,..,M]\}$

Ren-yuan Lyu
@ Chang Gung University

2013/10/29

$$\alpha_{s_t}(t) \equiv P(o_1,o_2....,o_t,s_t) \qquad = \sum_{s_{t-1}=1}^{N} \alpha_{s_{t-1}}(t-1)\cdot a_{s_{t-1}s_t}\cdot b_{s_t}(o_t)$$

$$\beta_{s_t}(t) \equiv P(o_{t+1},o_{t+2},...,o_{T-1},o_T \mid s_t) = \sum_{s_{t+1}=1}^{N} a_{s_t s_{t+1}}\cdot b_{s_{t+1}}(o_{t+1})\cdot \beta_{s_{t+1}}(t+1)$$

$$\eta_{s_{t-1}s_t}(t) \equiv P(s_{t-1},s_t,\underline{O}) \qquad = \alpha_{s_{t-1}}(t-1)\cdot a_{s_{t-1}s_t}\cdot b_{s_t}(o_t)\cdot \beta_{s_t}(t)$$

$$\eta_{s_t}(t) \equiv P(s_t,\underline{O}) = \sum_{s_{t-1}=1}^{N} \eta_{s_{t-1}s_t}(t) \qquad = \alpha_{s_t}(t)\cdot \beta_{s_t}(t)$$

$$P(\underline{O}) = \sum_{s_T=1}^{N} \alpha_{s_T}(T) = \beta_{s_0}(0) = \sum_{s_t=1}^{N} \eta_{s_t}(t)$$

$$\gamma_{s_{t-1}s_t}(t) \equiv P(s_{t-1},s_t \mid \underline{O}) = \eta_{s_{t-1}s_t}(t)/P(\underline{O})$$

$$\gamma_{s_t}(t) \equiv P(s_t \mid \underline{O}) = \eta_{s_t}(t)/P(\underline{O})$$

The training formula for c, μ, σ

$$\xi_{s_{t-1}s_t k_t}(t) \equiv P(s_{t-1},s_t,k_t,\underline{O}) \qquad = \alpha_{s_{t-1}}(t-1)\cdot a_{s_{t-1}s_t}\cdot c_{s_t}\cdot f_{s_t k_t}(o_t)\cdot \beta_{s_t}(t)$$

$$\xi_{s_t k_t}(t) \equiv P(s_t,k_t,\underline{O}) = \sum_{s_{t-1}=1}^{N} \xi_{s_{t-1}s_t k_t}(t)$$

$$\zeta_{s_t k_t}(t) \equiv P(s_t,k_t \mid \underline{O}) = \xi_{s_t k_t}(t)/P(\underline{O})$$

$$\zeta_{s_t}(t) \equiv P(s_t \mid \underline{O}) = \sum_{k_t=1}^{M} \zeta_{s_t k_t}(t) \overset{\text{尚未證明}}{=\!=\!=\!=\!=\!=\!=\!=} \gamma_{s_t}(t)$$

$$f_{s_t k_t}(o_t) \equiv f\left(o_t;\mu_{s_t k_t},\sigma^2_{s_t k_t}\right)$$

k_t : the mixture index at time t

$f\left(o;\mu,\sigma^2\right)$: the Gaussian distribution

$$\hat{c}_{jk} = \frac{<\zeta_{jk}(t)>}{<\zeta_j(t)>} = \frac{\frac{1}{T}\sum_{t=1}^{T}\zeta_{jk}(t)}{\frac{1}{T}\sum_{t=1}^{T}\zeta_j(t)} = \frac{\sum_{t=1}^{T}\zeta_{jk}(t)}{\sum_{t=1}^{T}\zeta_j(t)} =\!=\!=\!=\!=\!=\!=\!=\!=\!= \frac{\mathrm{Z}_{jk}}{\Gamma_j}$$

$$\hat{\mu}_{jk} = \frac{<\zeta_{jk}(t)\cdot o_t>}{<\zeta_{jk}(t)>} = \frac{\sum_{t=1}^{T}\zeta_{jk}(t)\cdot o_t}{\sum_{t=1}^{T}\zeta_{jk}(t)} =\!=\!=\!=\!=\!=\!=\!=\!=\!=\!=\!=\!=\!=\!= \frac{\mathrm{M}_{jk}}{\mathrm{Z}_{jk}}$$

$$\hat{\sigma}^2_{jk} = \frac{<\zeta_{jk}(t)\cdot\left(o_t-\hat{\mu}_{jk}\right)^2>}{<\zeta_{jk}(t)>} = \frac{\sum_{t=1}^{T}\zeta_{jk}(t)\cdot\left(o_t-\hat{\mu}_{jk}\right)^2}{\sum_{t=1}^{T}\zeta_{jk}(t)} =\!=\!=\!= \frac{U_{jk}}{\mathrm{Z}_{jk}}$$

Ren-yuan Lyu
@ Chang Gung University

The training algorithm for c, μ, σ

$$\Gamma_i = \sum_{t=0}^{T-1} \gamma_i(t)$$

$$\Gamma_{ij} = \sum_{t=1}^{T} \gamma_{ij}(t)$$

$$\Gamma_0 = \gamma_0(0)$$

$$\Gamma_{0j} = \gamma_{0j}(1)$$

$$\Gamma_j = \sum_{t=1}^{T} \gamma_j(t)$$

$$\Delta_{jo} = \sum_{t=1}^{T} \gamma_j(t) \cdot \delta(o_t - o)$$

$$\hat{\pi}_j = \hat{a}_{0j}\,|_{\underline{O}} = \frac{\Gamma_{0j}}{\Gamma_0}$$

$$\hat{a}_{ij}\,|_{\underline{O}} = \frac{\Gamma_{ij}}{\Gamma_i}$$

$$\hat{b}_j(o)\,|_{\underline{O}} = \frac{\Delta_{jo}}{\Gamma_j}$$

$$\boxed{\begin{aligned}
&\mathrm{Z}_{jk} = \sum_{t=1}^{T} \zeta_{jk}(t) \\
&\Rightarrow \hat{c}_{jk} = \frac{\mathrm{Z}_{jk}}{\Gamma_j} \\
&\mathrm{M}_{jk} = \sum_{t=1}^{T} \zeta_{jk}(t) \cdot o_t \\
&\Rightarrow \hat{\mu}_{jk} = \frac{\mathrm{M}_{jk}}{\mathrm{Z}_{jk}} \\
&V_{jk} = \sum_{t=1}^{T} \zeta_{jk}(t) \cdot \left(o_t - \hat{\mu}_{jk}\right)^2 \\
&\Rightarrow \hat{\sigma}_{jk}^2 = \frac{V_{jk}}{\mathrm{Z}_{jk}}
\end{aligned}}$$

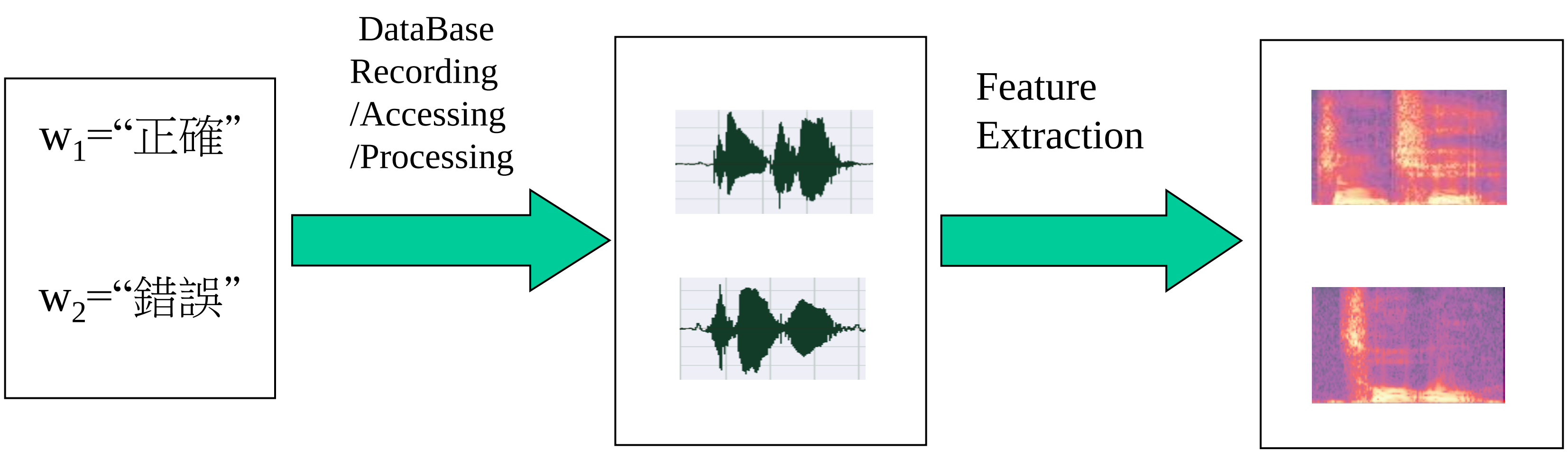
For multi-dimensional observation vector

$$\boxed{\begin{aligned}
&\vec{\mathrm{M}}_{jk} = \sum_{t=1}^{T} \zeta_{jk}(t) \cdot \vec{o}_t \\
&\Rightarrow \hat{\vec{\mu}}_{jk} = \frac{\vec{\mathrm{M}}_{jk}}{\mathrm{Z}_{jk}} \\
&\underline{\underline{V}}_{jk} = \sum_{t=1}^{T} \zeta_{jk}(t) \cdot \left(\vec{o}_t - \hat{\vec{\mu}}_{jk}\right) \cdot \left(\vec{o}_t - \hat{\vec{\mu}}_{jk}\right)^T \\
&\Rightarrow \underline{\underline{U}}_{jk} = \frac{\underline{\underline{V}}_{jk}}{\mathrm{Z}_{jk}}
\end{aligned}}$$

Ren-yuan Lyu
@ Chang Gung University

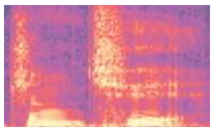
Isolated Word Recognition using CHMM

- Data Preparation



- Training

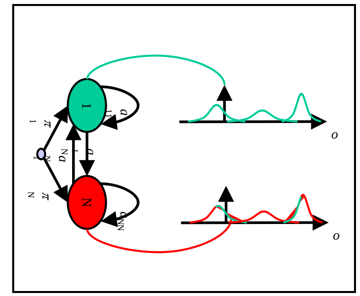
$\underline{O}$ for w_1



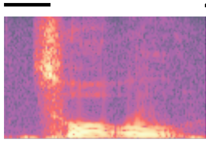
Training

Make $P(O|\pi_1,A_1,B_1)$
As large as possible

HMM_1 for w_1



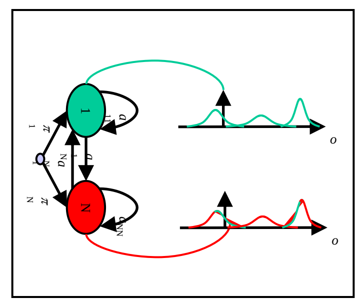
$\underline{O}$ for w_2



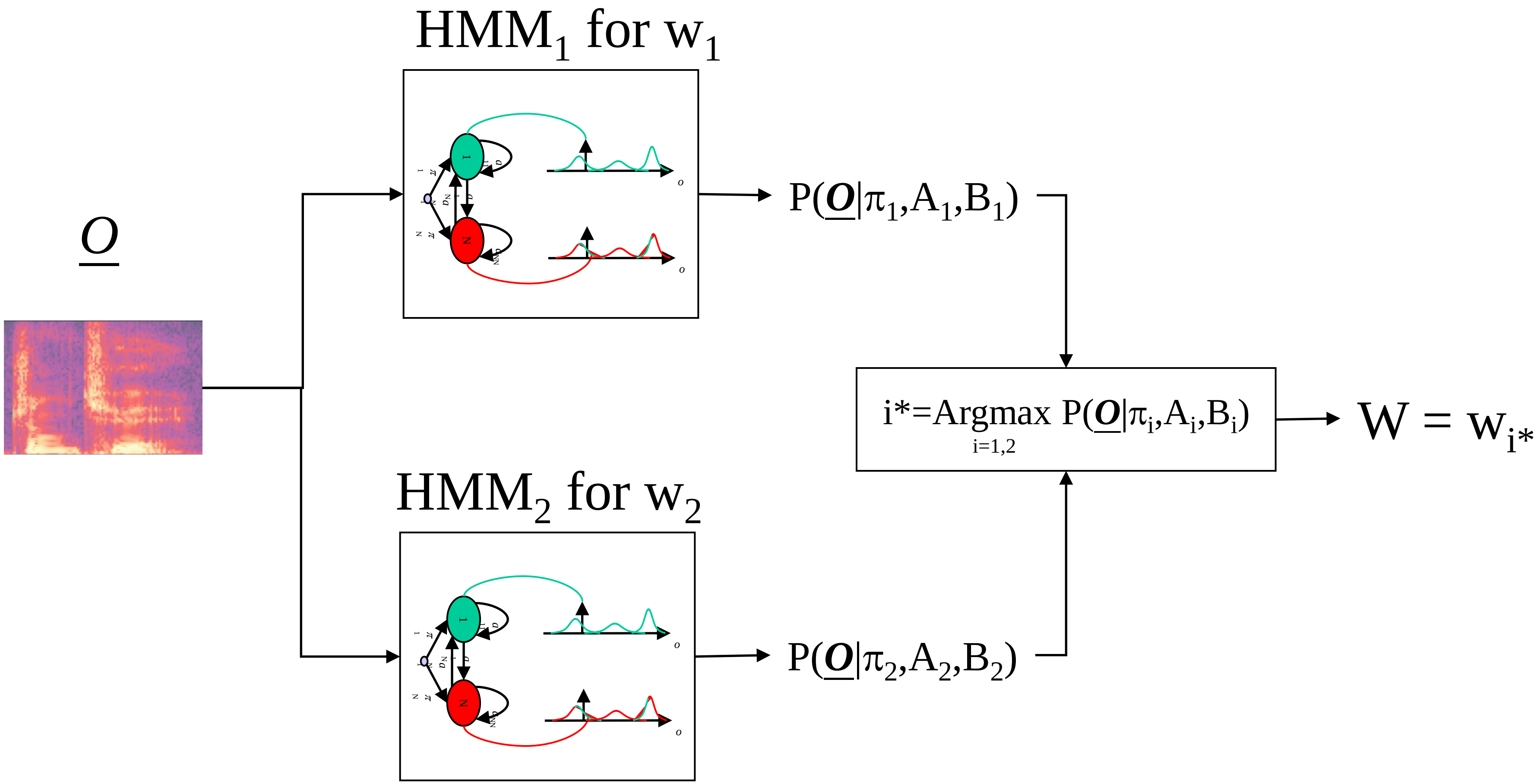
Training

Make $P(O|\pi_2,A_2,B_2)$
As large as possible

HMM_2 for w_2

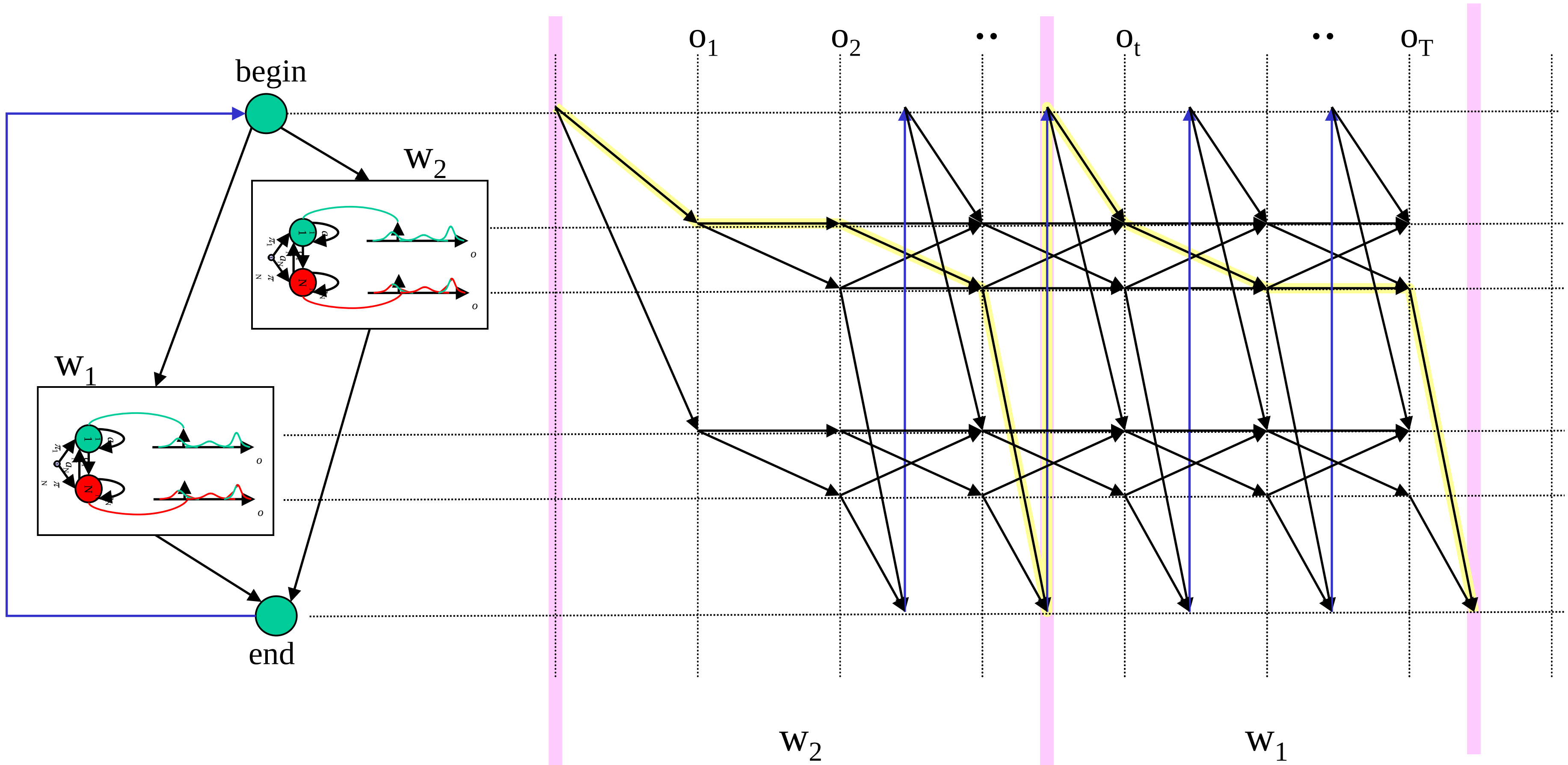


- Recognition



Ren-yuan Lyu
@ Chang Gung University

Continuous Speech Recognition



Ren-yuan Lyu
@ Chang Gung University

Large Vocabulary Speech Recognition

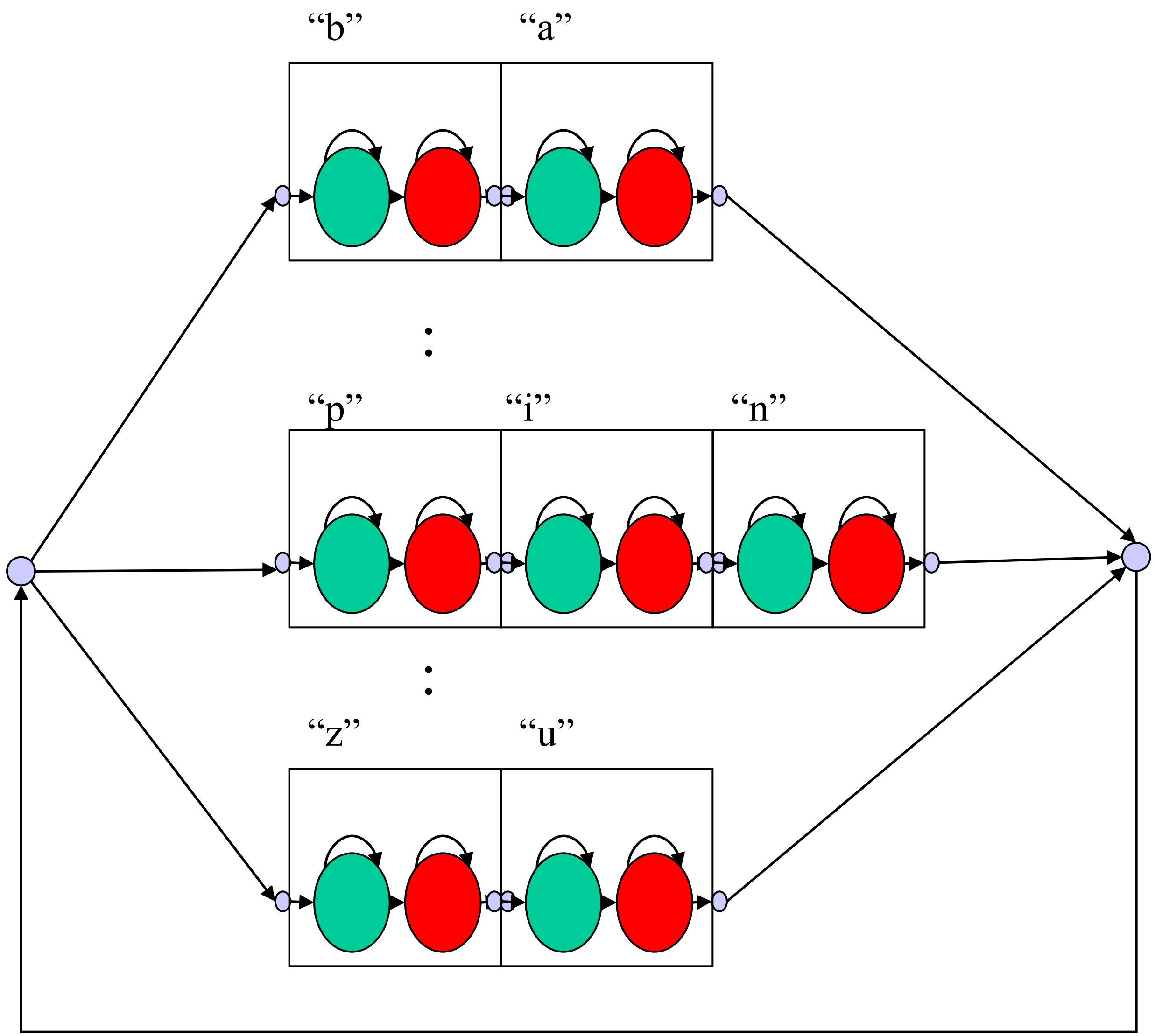
- Sentence = “今天,天氣,不錯”
- Sentence = W = $W_1W_2\ldots W_{(Nw)}$
  - “今天,天氣,不錯” = “今天” , ”天氣” , ”不錯”
- Word, W=C= $C_1C_2\ldots C_{(Nc)}$
  - “今天” = “今” , ”天”
- Character, C = S = $S_1S_2\ldots S_{(Ns)}$
  - “今” = “zin”
- Syllable, S = P =$P_1P_2\ldots P_{(Np)}$
  - “zin” = “z”, “i”, “n”
- Phone, P, has some variations
  - mono-phone,
    - “**z**”, “**i**’, “**n**”
  - bi-phone,
    - “**z**+i”, “**i**+n’, “**n**+sil”
  - tri-phone,
    - “sil-**z**+i”, “z-**i**+n’, “i-**n**+sil”
  - Initial/Final
    - “**z**”, “**in**”
    - “**z**+i”, “**in**”
    - “**z**+in”, “**in**”

要用1個HMM來代表何層次的語言單位？

N(Sentence) = ∞
N(Word) = 100K
N(Character) = 10K
N(Syllable) = 1K
N(Phone) =
N(Mono-phone) = .1K
N(bi-phone) = .5K
N(tri-phone) = 1 K
N(Initial/Final) = .5K

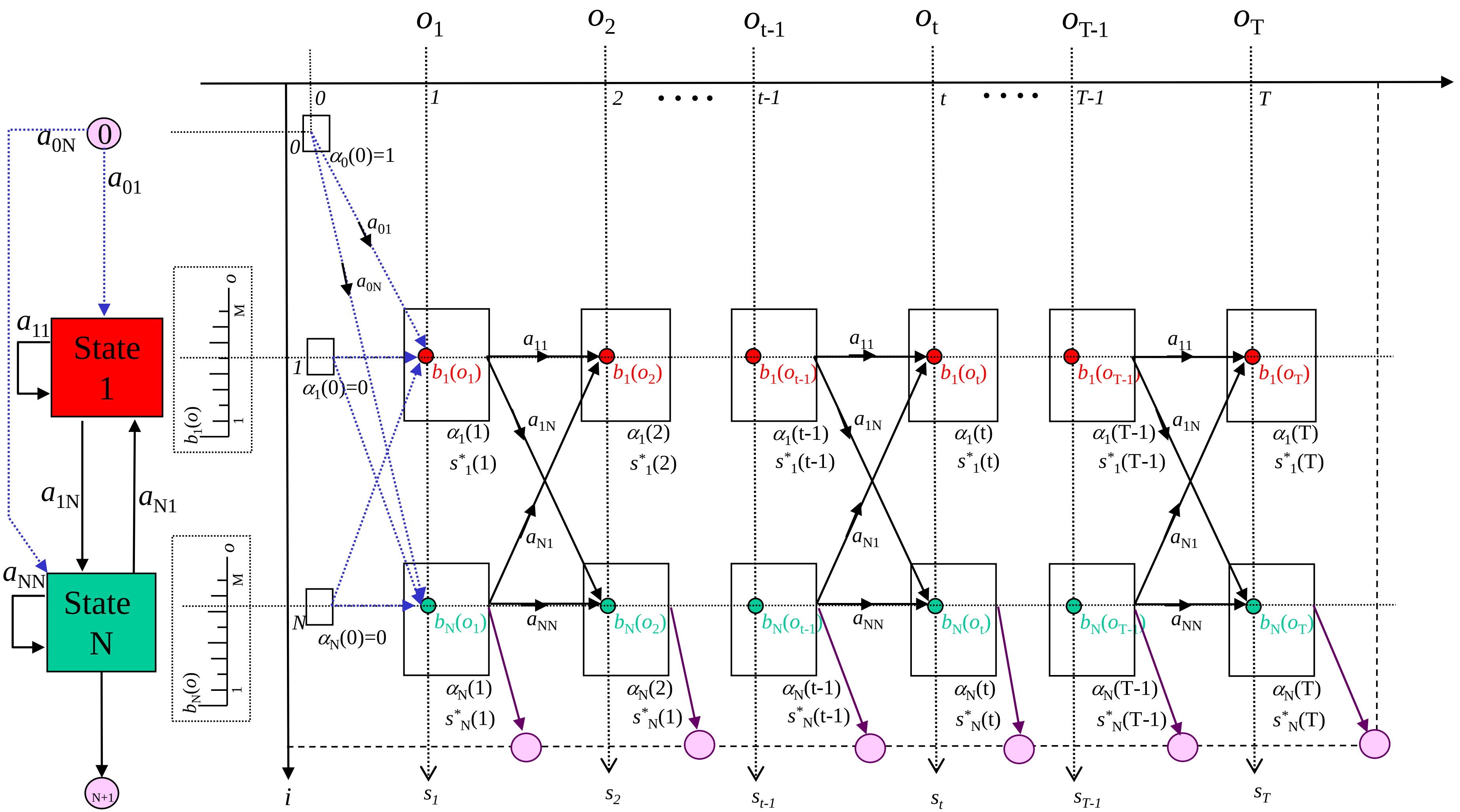
Ren-yuan Lyu
@ Chang Gung University

Continuous Syllable Recognition



Ren-yuan Lyu
@ Chang Gung University

Review of Viterbi Algorithm in HMM



$$\alpha_j(0) = \begin{cases} 1, & j=1 \\ 0, & j\neq 1,\ j\in[1..N] \end{cases}$$

$$s_j^*(t) = i^* = \underset{i\in[1..N]}{ArgMax}\left(\alpha_i(t-1)\cdot a_{ij}\right) \quad , j\in[1..N]$$

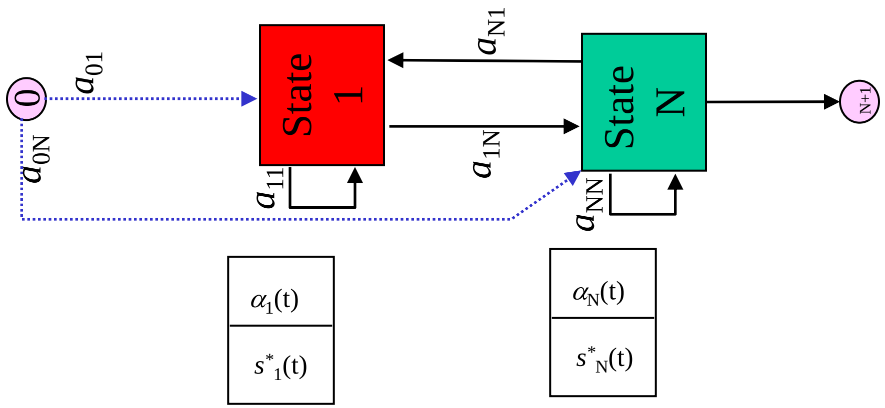
$$\alpha_j(t) = \left(\alpha_{i^*}(t-1)\cdot a_{i^*j}\right)\cdot b_j(o_t) \qquad , j\in[1..N]$$

$$s_{N+1}^*(t) = N$$

$$\alpha_{N+1}(t) = \left(\alpha_N(t)\cdot a_{N(N+1)}\right)$$

Ren-yuan Lyu
@ Chang Gung University

Token Passing

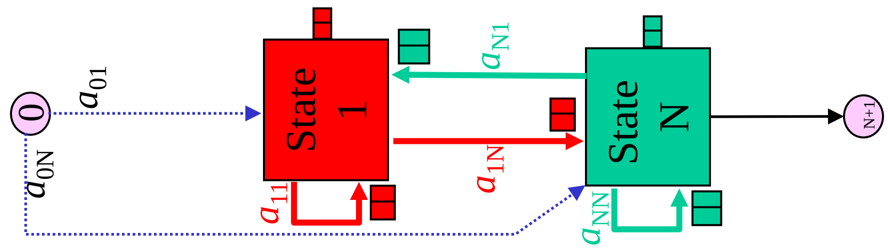


$$s_j^*(t) = i^* = \underset{i \in [1..N]}{ArgMax}\left(\alpha_i(t-1) \cdot a_{ij}\right) \quad , j \in [1..N]$$

$$\alpha_j(t) = \left(\alpha_{i^*}(t-1) \cdot a_{i^* j}\right) \cdot b_j(o_t) \quad , j \in [1..N]$$

For each state i,

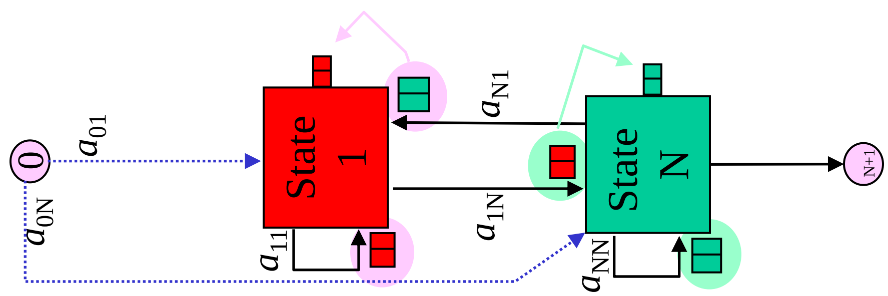
Passing the token of state i to all its connecting state j



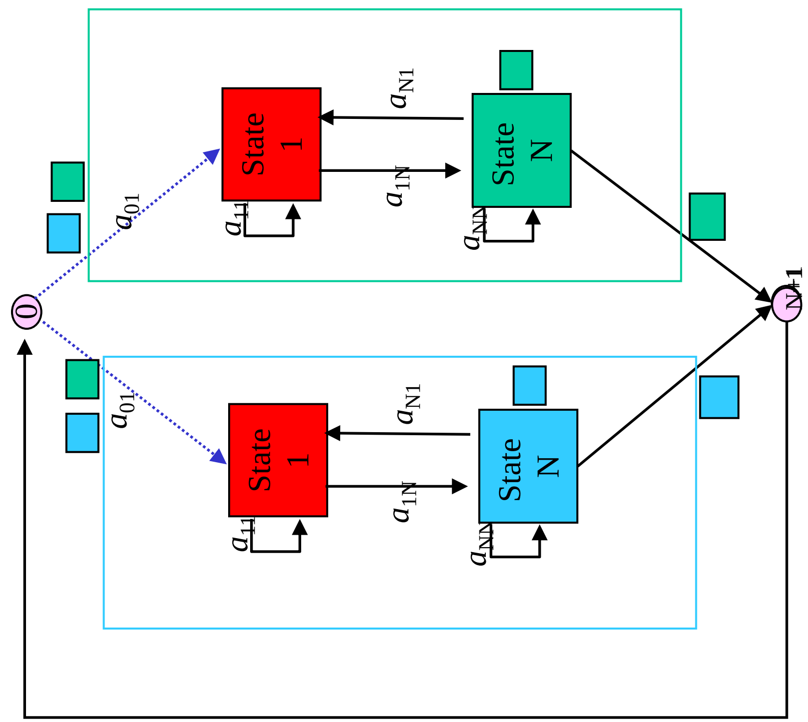
For each state j,

Find the best token of all tokens which are passed to state j,

Then update it as the new token of state j



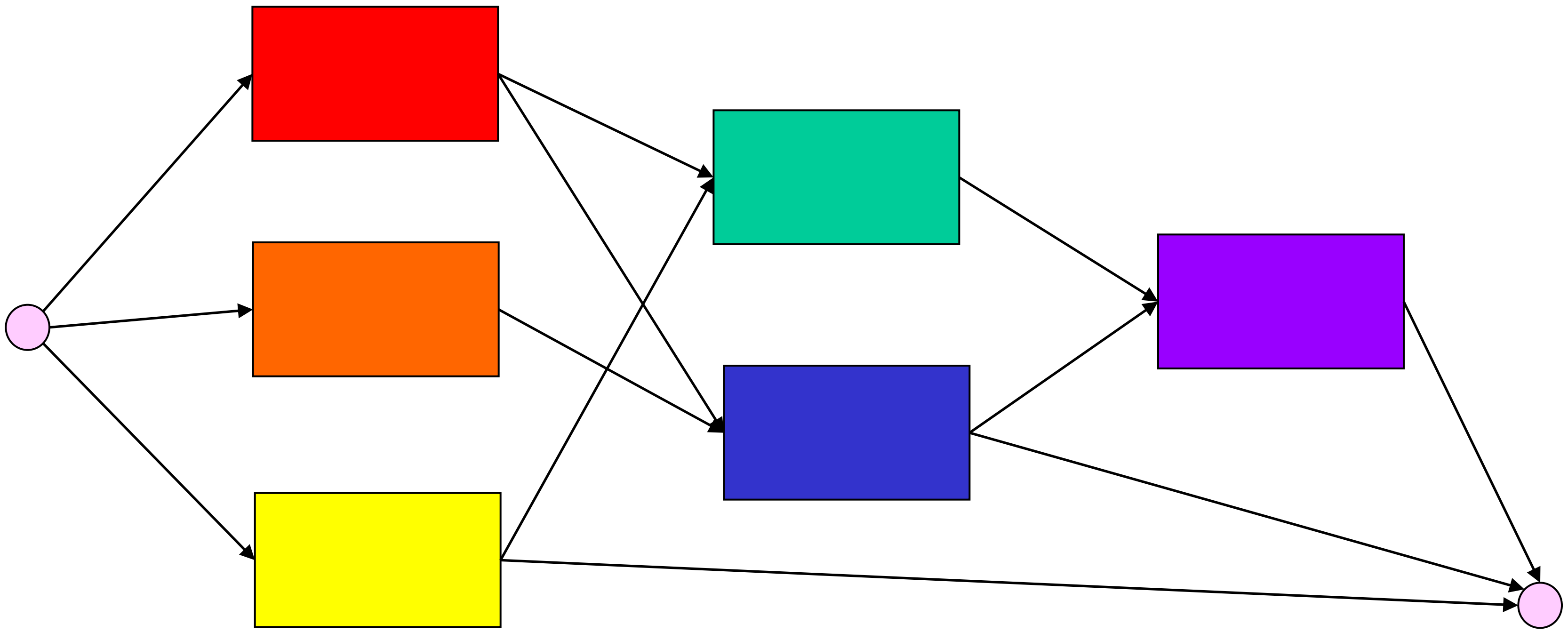
Ren-yuan Lyu
@ Chang Gung University



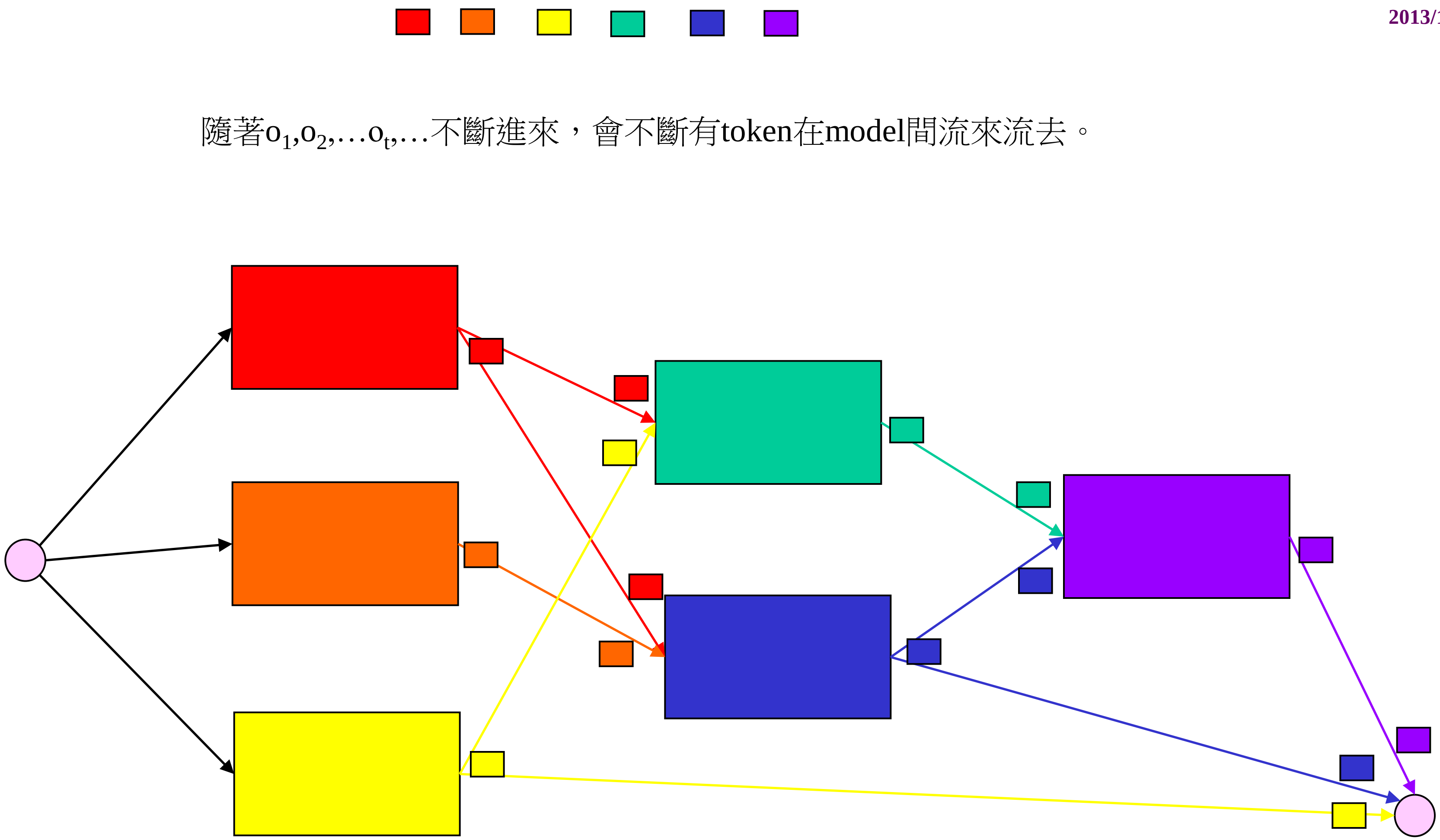
這是一個動態的圖，隨著t的前進，
token 會不斷從各個model框框流出，
再進入各個model，
歷經最佳分數選取的機制，再Update目前token後，
繼續流出…

Ren-yuan Lyu
@ Chang Gung University

假設根據「文法規則」，model之間可以如下圖連接。



隨著$o_1,o_2,\ldots o_t,\ldots$不斷進來，會不斷有token在model間流來流去。



在任意時間點t,這些token會不斷從每個model流出，並流入連接的下一個model，經歷選擇最佳token，並經過model的「消化」後，產生新的token，再流出。

每一個token (at time t, for model w)記載著到時間t為止，