NLP

Introduction to NLP

Hidden Markov Models

Markov Models

- Sequence of random variables that aren't independent
- Examples
  - weather reports
  - text

Properties

- Limited horizon:

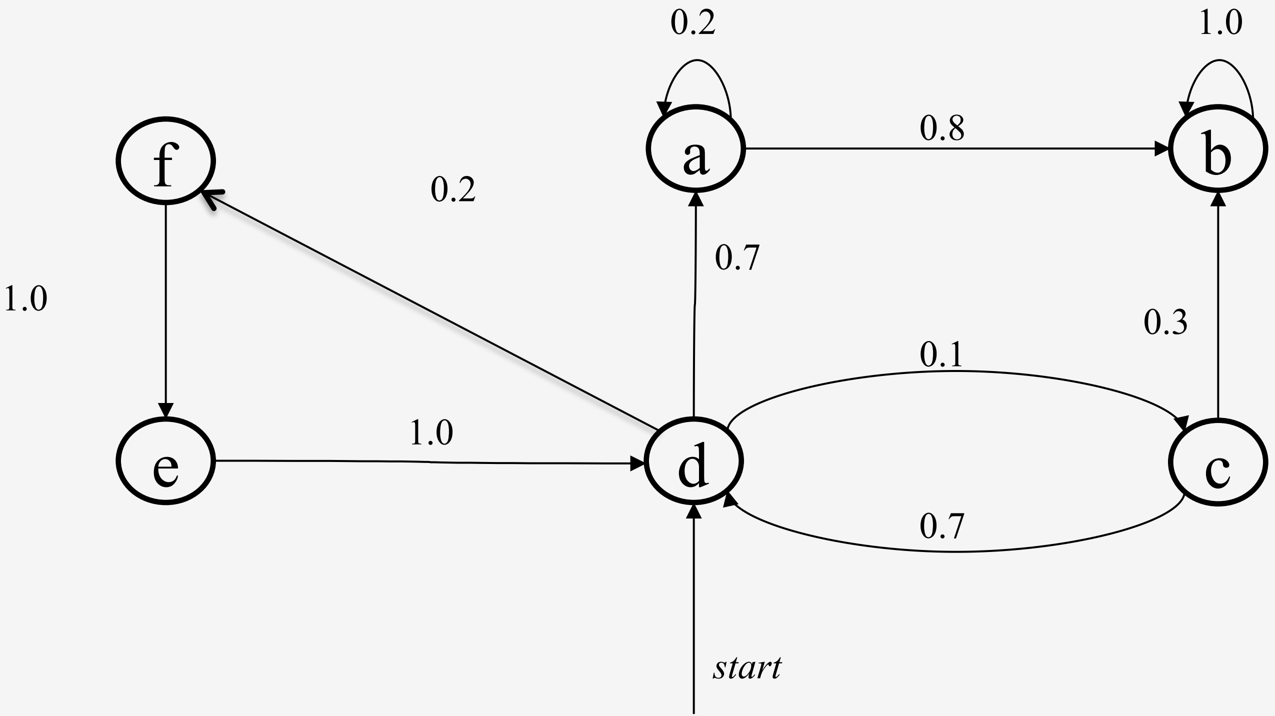
  $$P(X_{t+1} = s_k | X_1, \ldots, X_t) = P(X_{t+1} = s_k | X_t)$$

- Time invariant (stationary)

  $$= P(X_2 = s_k | X_1)$$

- Definition: in terms of a transition matrix A and initial state probabilities Π.

Example



Visible MM

$$P(X_1,\ldots X_T) = P(X_1)\, P(X_2|X_1)\, P(X_3|X_1,X_2)\, \ldots\, P(X_T|X_1,\ldots,X_{T-1})$$

$$= P(X_1)\, P(X_2|X_1)\, P(X_3|X_2)\, \ldots\, P(X_T|X_{T-1})$$

$$= \pi_{X_1} \prod_{t=1}^{T-1} a_{X_t X_{t+1}}$$

$$P(d, a, b) = P(X_1{=}d)\, P(X_2{=}a|X_1{=}d)\, P(X_3{=}b|X_2{=}a)$$

$$= 1.0 \times 0.7 \times 0.8$$

$$= 0.56$$

Hidden MM

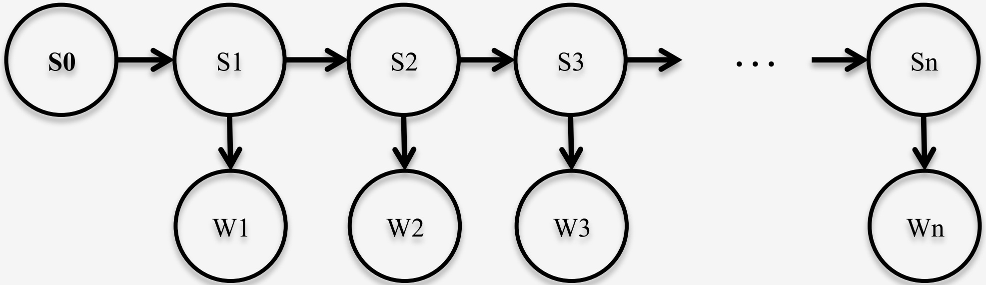
- Motivation
  - Observing a sequence of symbols
  - The sequence of states that led to the generation of the symbols is hidden
- Definition
  - Q = sequence of states
  - O = sequence of observations, drawn from a vocabulary
  - q_0,q_f = special (start, final) states
  - A = state transition probabilities
  - B = symbol emission probabilities
  - Π = initial state probabilities
  - $\mu = (A,B,\Pi)$ = complete probabilistic model

Hidden MM

- Uses
  - part of speech tagging
  - speech recognition
  - gene sequencing

Hidden Markov Model (HMM)

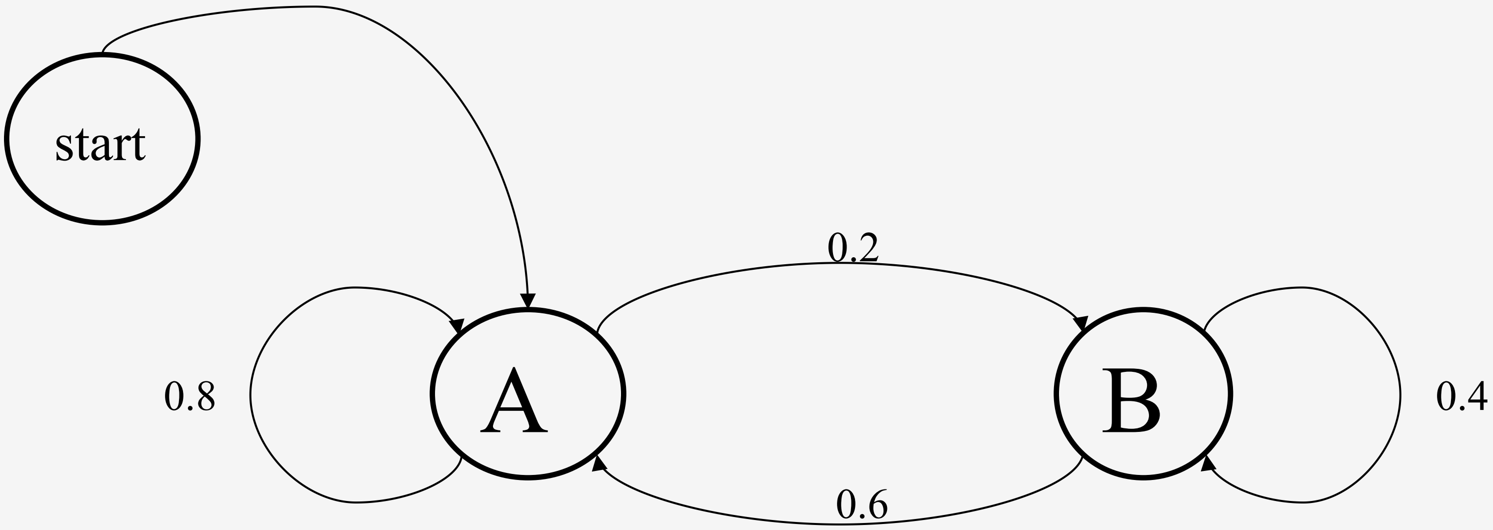
- Can be used to model state sequences and observation sequences
- Example:
  - $P(\mathbf{s},\mathbf{w}) = \prod_i P(s_i|s_{i-1})P(w_i|s_i)$



Generative Algorithm

- Pick start state from Π
- For t = 1..T
  - Move to another state based on A
  - Emit an observation based on B

State Transition Probabilities



Emission Probabilities

- $P(O_t=k|X_t=s_i,X_t+1=s_j) = b_{ijk}$

	x	y	z
A	0.7	0.2	0.1
B	0.3	0.5	0.2

All Parameters of the Model

- Initial
  - P(A|start) = 1.0, P(B|start) = 0.0
- Transition
  - P(A|A) = 0.8, P(A|B) = 0.6, P(B|A) = 0.2, P(B|B) = 0.4
- Emission
  - P(x|A) = 0.7, P(y|A) = 0.2, P(z|A) = 0.1
  - P(x|B) = 0.3, P(y|B) = 0.5, P(z|B) = 0.2

Observation Sequence "yz"

- Starting in state A, P(yz) = ?
- Possible sequences of states:
  - AA
  - AB
  - BA
  - BB
- P(yz) = P(yz|AA) + P(yz|AB) + P(yz|BA) + P(yz|BB) =

  = .8 x .2 x .8 x .1

  \+ .8 x .2 x .2 x .2

  \+ .2 x .5 x .4 x .2

  \+ .2 x .5 x .6 x .1

  = .0128+.0064+.0080+.0060 =.0332

States and Transitions

- The states encode the most recent history
- The transitions encode likely sequences of states
  - e.g., Adj-Noun or Noun-Verb
  - or perhaps Art-Adj-Noun
- Use MLE to estimate the transition probabilities

Emissions

- Estimating the emission probabilities
  - Harder than transition probabilities
  - There may be novel uses of Word/POS combinations
- Suggestions
  - It is possible to use standard smoothing
  - As well as heuristics (e.g., based on the spelling of the words)

Sequence of Observations

- The observer can only see the emitted symbols
- Observation likelihood
  - Given the observation sequence S and the model $\mu = (A,B,\Pi)$, what is the probability $P(S|\mu)$ that the sequence was generated by that model.
- Being able to compute the probability of the observations sequence turns the HMM into a language model

Tasks With HMM

- Tasks
  - Given $\mu = (A,B,\Pi)$, find $P(O|\mu)$
  - Given O, μ, what is $(X_1,\ldots X_{T+1})$
  - Given O and a space of all possible μ, find model that best describes the observations
- Decoding – most likely sequence
  - tag each token with a label
- Observation likelihood
  - classify sequences
- Learning
  - train models to fit empirical data

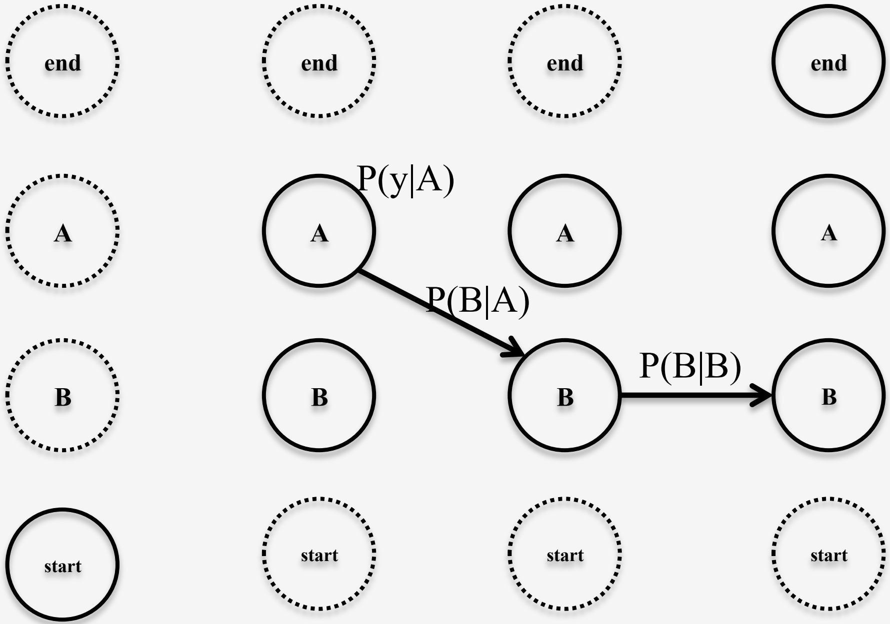
Inference

- Find the most likely sequence of tags, given the sequence of words
  - $t^* = \text{argmax}_t\ P(t|w)$
- Given the model μ, it is possible to compute P (t|w) for all values of t
- In practice, there are way too many combinations
- Possible solution:
  - Use beam search (partial hypotheses)
  - At each state, only keep the k best hypotheses so far
  - May not work

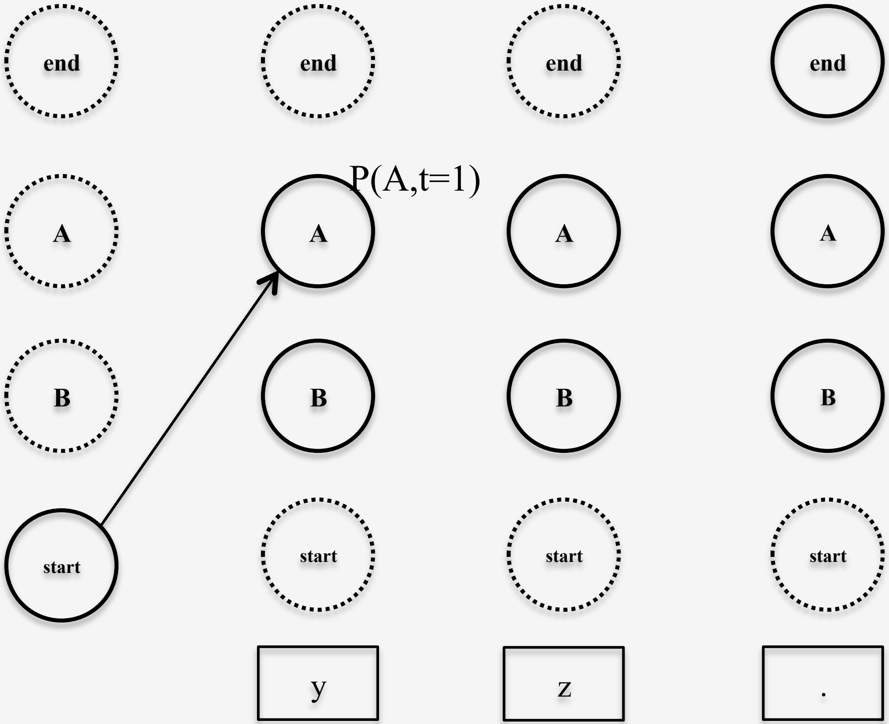
Viterbi Algorithm

- Find the best path up to observation i and state s
- Characteristics
  - Uses dynamic programming
  - Memoization
  - Backpointers

HMM Trellis

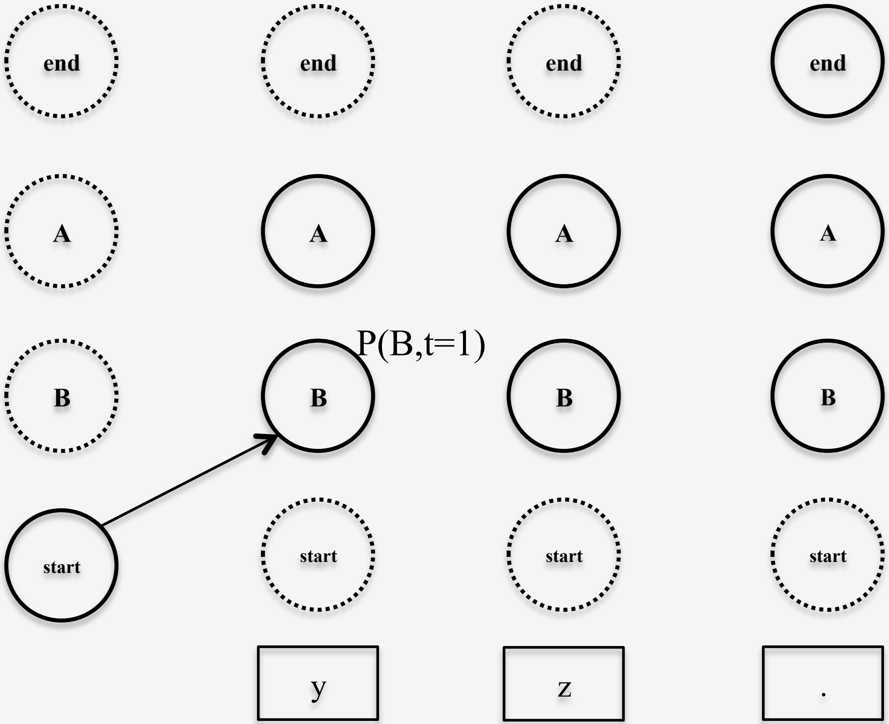


HMM Trellis



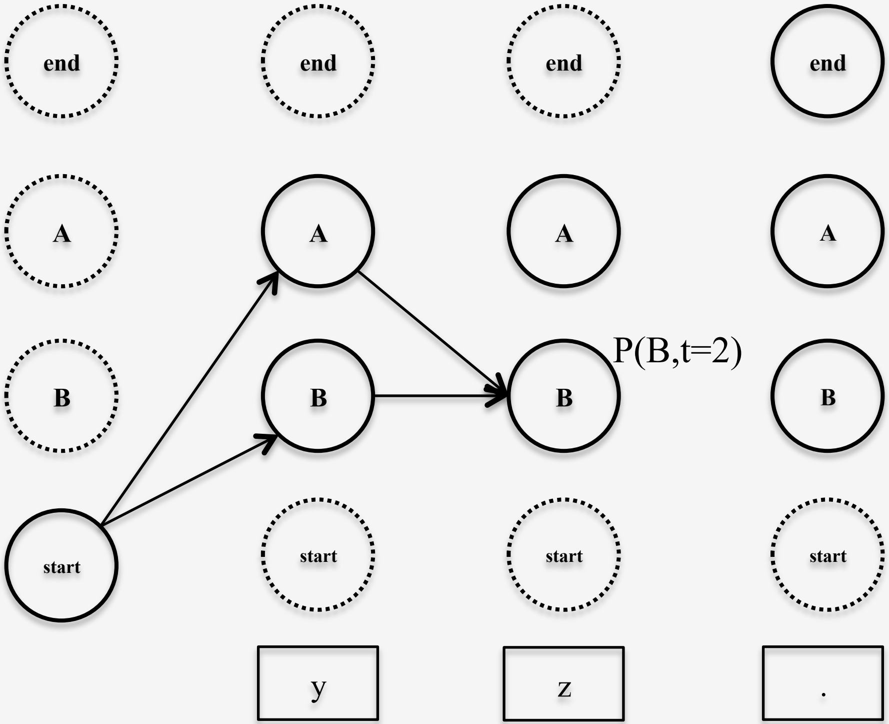
$P(A,t=1) =$
$P(start) \times P(A|start) \times P(y|A)$

HMM Trellis



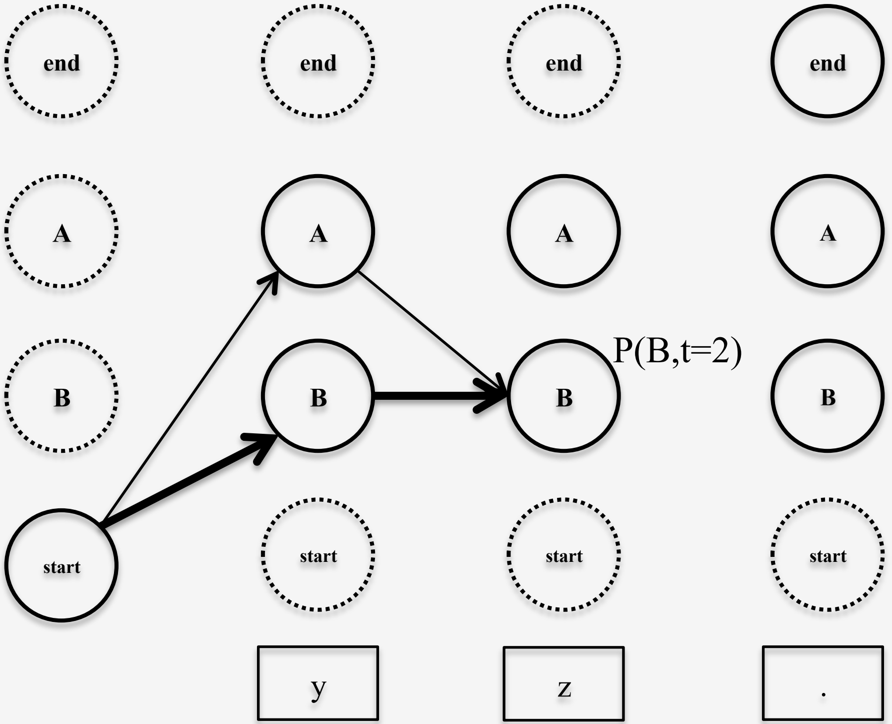
$P(B,t=1) =$
$P(start) \times P(B|start) \times P(y|B)$

HMM Trellis

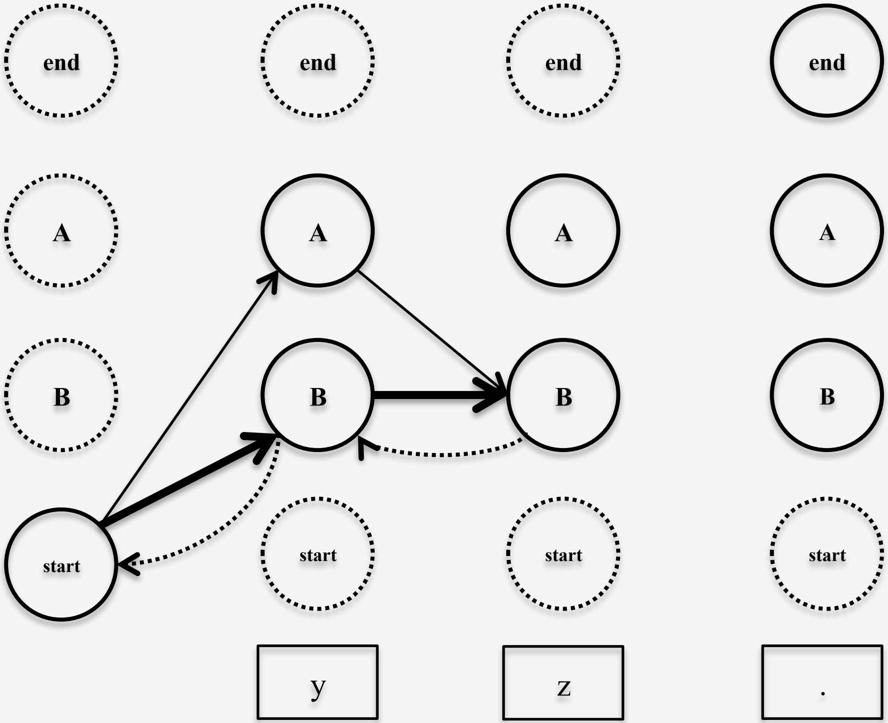


$P(B,t=2) =$
$\max\ (P(A,t=1) \times P(B|A) \times P(z|b),$
$P(B,t=1) \times P(B|B) \times P(z|b))$

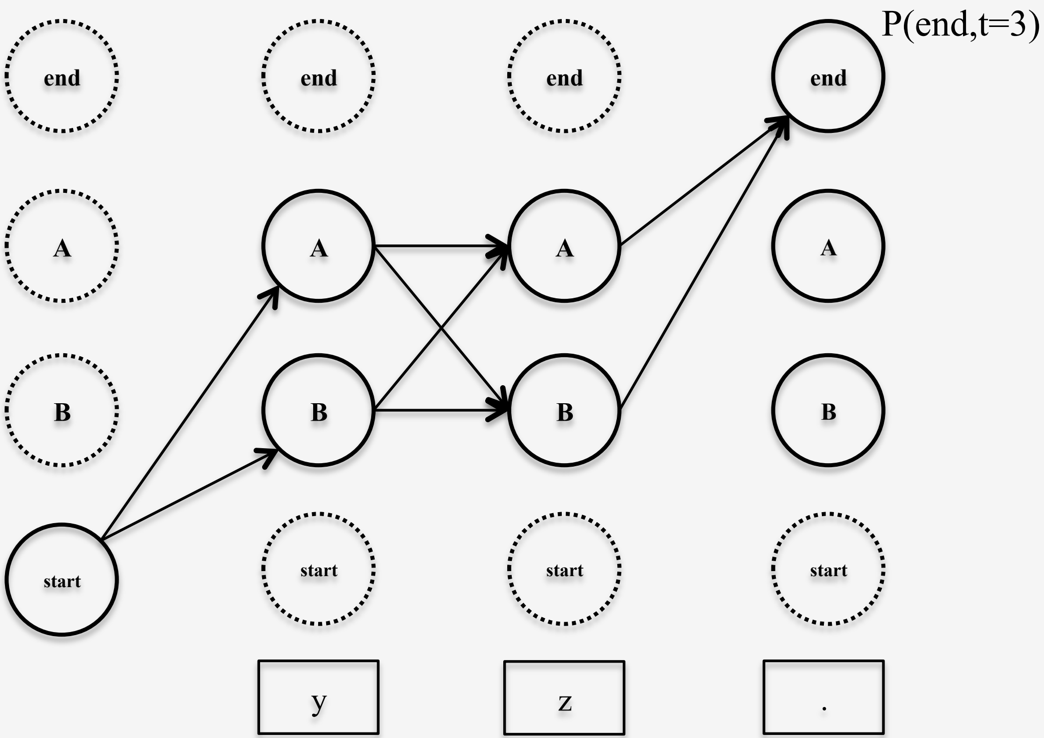
HMM Trellis



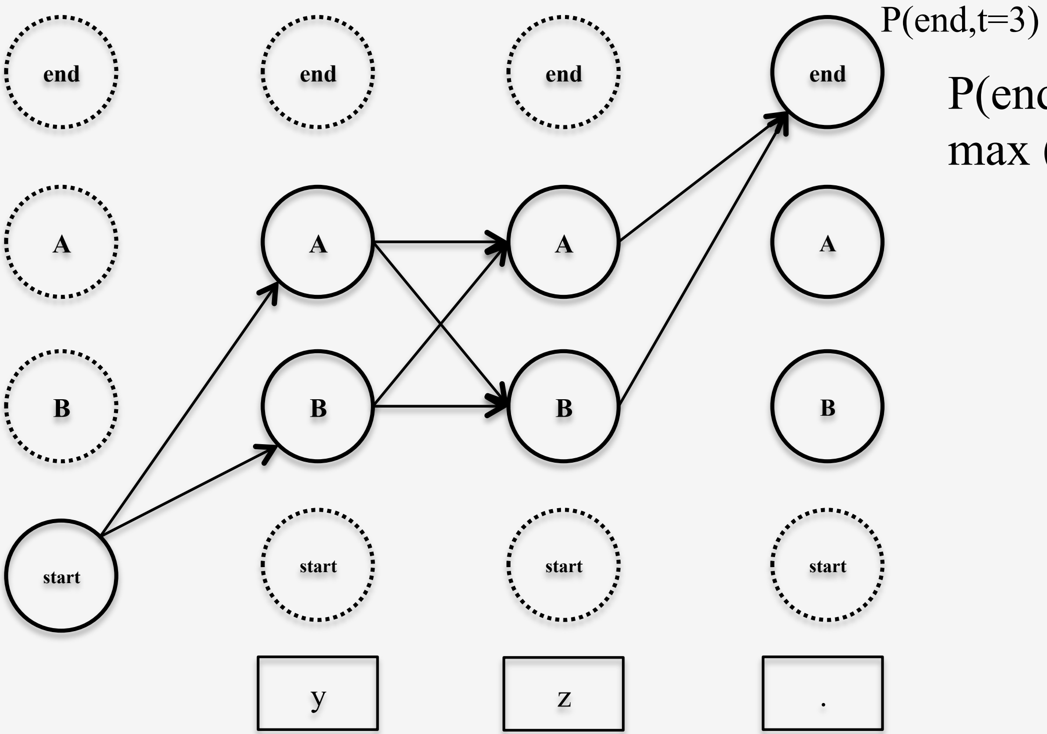
HMM Trellis



HMM Trellis

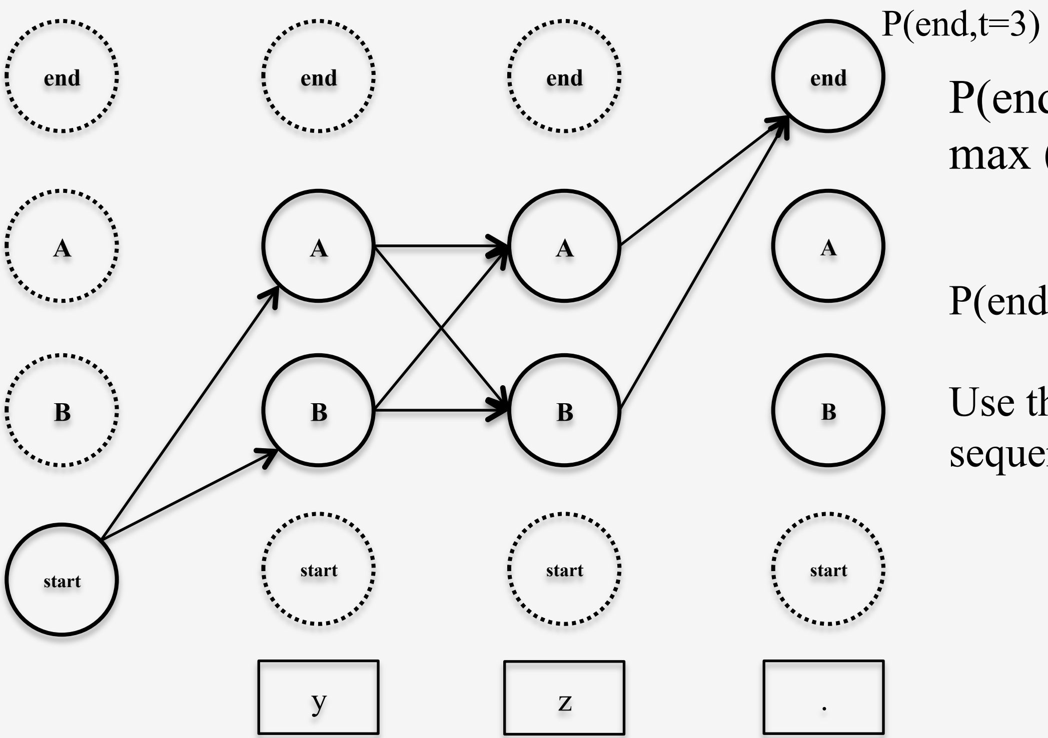


HMM Trellis



P(end,t=3) =
max (P(A,t=2) x P(end|A),
P(B,t=2) x P(end|B))

HMM Trellis



P(end,t=3) =
max (P(A,t=2) x P(end|A),
P(B,t=2) x P(end|B))

P(end,t=3) = best score for the sequence

Use the backpointers to find the sequence of states.

NLP