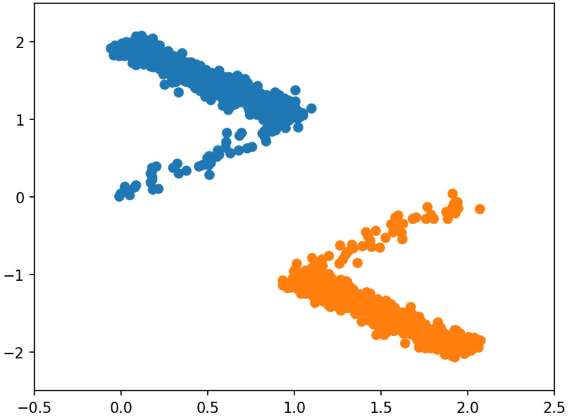
Discriminative Training

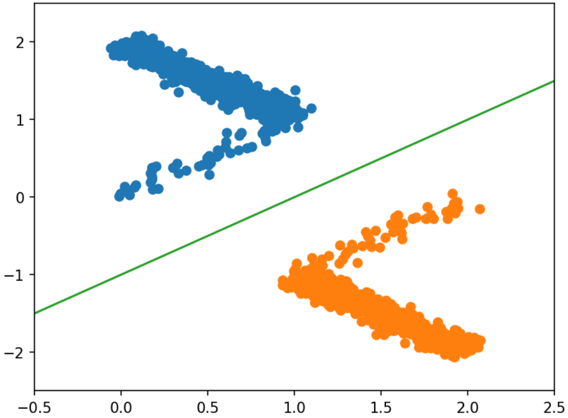
Hao Tang

Automatic Speech Recognition—ASR Lecture 14
10 March 2022

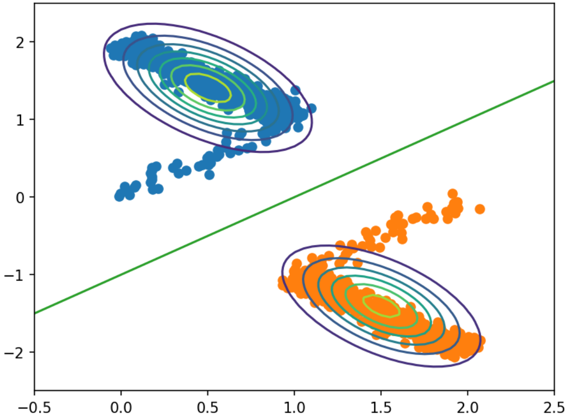
Discriminative vs Generative Training



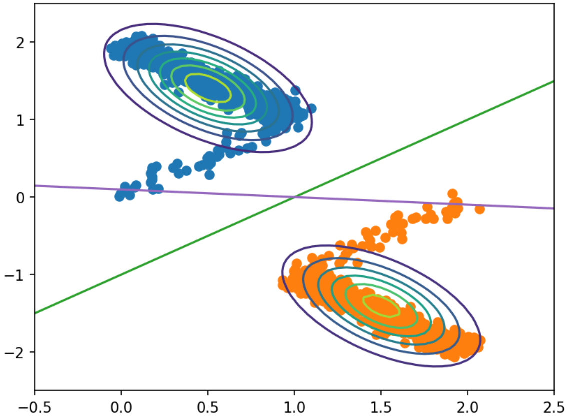
Discriminative vs Generative Training



Discriminative vs Generative Training



Discriminative vs Generative Training



Discriminative vs Generative

- It is not about whether one is better than the other.

Discriminative vs Generative

- It is not about whether one is better than the other.
- Should we use the samples (and computation) to learn the decision boundary or the data distribution?

Discriminative vs Generative

- It is not about whether one is better than the other.
- Should we use the samples (and computation) to learn the decision boundary or the data distribution?
- The discriminative approach might be a better solution when the boundary is simple to learn.

Discriminative vs Generative

- It is not about whether one is better than the other.
- Should we use the samples (and computation) to learn the decision boundary or the data distribution?
- The discriminative approach might be a better solution when the boundary is simple to learn.
- If the goal is to do prediction, we should focus on learning the bounary.

Recap of HMM Training

- Map words to a sequence of phones

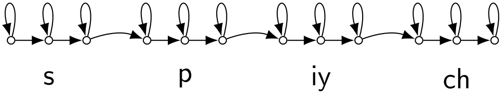
$$\text{speech} \rightarrow \text{s p iy ch}$$

Recap of HMM Training

- Map words to a sequence of phones

$$\text{speech} \rightarrow \text{s p iy ch}$$

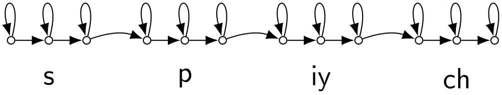
- Chain phone HMMs



- Map words to a sequence of phones

$$\text{speech} \rightarrow \text{s p iy ch}$$

- Chain phone HMMs



- Find parameters that maximize $p(X)$

Recap of HMM Training

- $p(X)$ really should be $p(X|W)$.

Recap of HMM Training

- $p(X)$ really should be $p(X|W)$.
- $p(X, Z)$ really should be $p(X, Z|W)$.

Recap of HMM Training

- $p(X)$ really should be $p(X|W)$.
- $p(X, Z)$ really should be $p(X, Z|W)$.
- Z is a valid sequence for W if the phone sequence produced by Z is the pronunciation of W.

Recap of HMM Training

- $p(X)$ really should be $p(X|W)$.
- $p(X, Z)$ really should be $p(X, Z|W)$.
- Z is a valid sequence for W if the phone sequence produced by Z is the pronunciation of W.
- s1 s1 s2 s2 s2 s3 s3 p1 p2 p3 iy1 iy1 iy1 iy2 iy2 iy2 iy3 iy3 iy3 ch1 ch2 ch3 is a valid sequence for the word "speech."

Recap of HMM Training

- $p(X, Z|W) = 0$ when Z is not a valid state sequence for W.

Recap of HMM Training

- $p(X, Z|W) = 0$ when Z is not a valid state sequence for W.
- $p(X, Z|W) = p(z_1)p(x_1|z_1) \prod_{t=2}^{T} p(z_t|z_{t-1})p(x_t|z_t)$ when Z is a valid state sequence for W.

Recap of HMM Training

- $p(X, Z|W) = 0$ when Z is not a valid state sequence for W.
- $p(X, Z|W) = p(z_1)p(x_1|z_1) \prod_{t=2}^{T} p(z_t|z_{t-1})p(x_t|z_t)$ when Z is a valid state sequence for W.
- Use $B(W)$ to denote the set of valid state sequences for W.

Recap of HMM Training

- $p(X, Z|W) = 0$ when Z is not a valid state sequence for W.
- $p(X, Z|W) = p(z_1)p(x_1|z_1)\prod_{t=2}^{T} p(z_t|z_{t-1})p(x_t|z_t)$ when Z is a valid state sequence for W.
- Use $B(W)$ to denote the set of valid state sequences for W.
- $p(X|W) = \sum_{Z \in B(W)} p(X, Z|W)$

Recap of HMM Training

- $\text{argmax}_\lambda\, p(X|W)$ can be solved with EM or gradient descent.
- $\text{argmax}_\lambda\, p(X|W)$ is a generative approach.
- The discriminative approach solves $\text{argmax}_\lambda\, p(W|X)$.

Maximum Mutual Information (MMI) (Bahl *et al.*, 1986)

$$p(W|X) = \frac{p(X|W)p(W)}{p(X)} = \frac{p(X|W)p(W)}{\sum_{W'} p(X|W')p(W')}$$

Maximum Mutual Information (MMI) (Bahl *et al.*, 1986)

$$p(W|X) = \frac{p(X|W)p(W)}{p(X)} = \frac{p(X|W)p(W)}{\sum_{W'} p(X|W')p(W')}$$

- How to compute the numerator $p(X|W)p(W)$?
- How to compute the denominator $\sum_{W'} p(X|W')p(W')$?
- Why is this called maximum mutual information (MMI)?

Numerator

$$p(X|W)p(W)$$

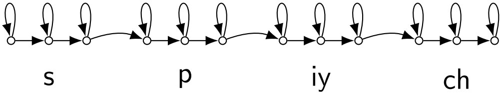
Numerator

$$p(X|W)p(W)$$

- Map words to a sequence of phones

$$\text{speech} \rightarrow \text{s p iy ch}$$

- Chain phone HMMs



- Compute $p(X|W)$
- Compute $p(W)$ with a language model

Denominator

- It's computationally expensive to compute the denominator exactly.

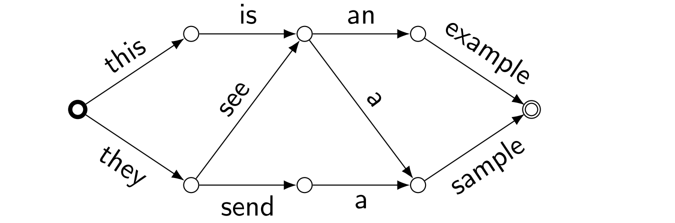
$$\sum_{W'} p(X|W')p(W')$$

- Instead we can approximate it with a set of high-probability word sequences D.

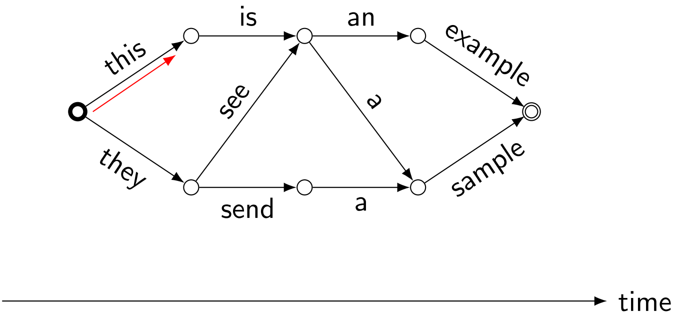
$$\sum_{W' \in D} p(X|W')p(W')$$

- The set of high-probability sequences D is called a **lattice**.

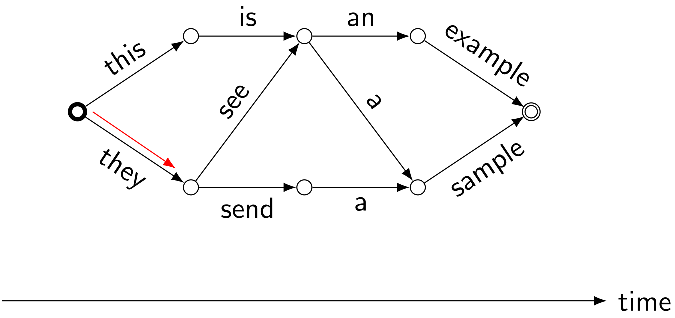
Lattice



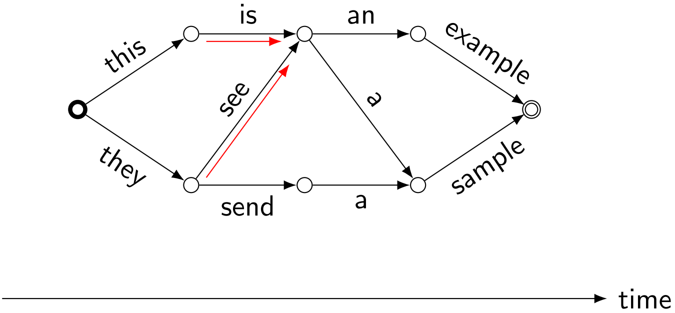
Lattice



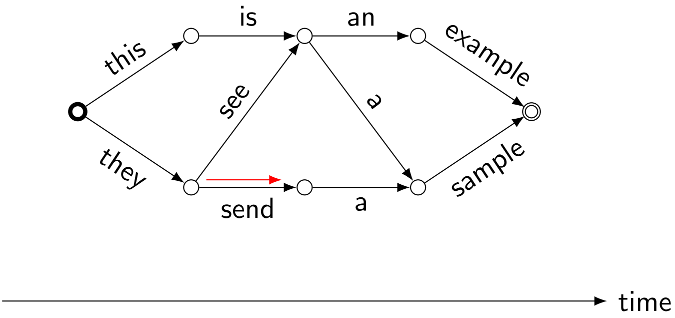
Lattice



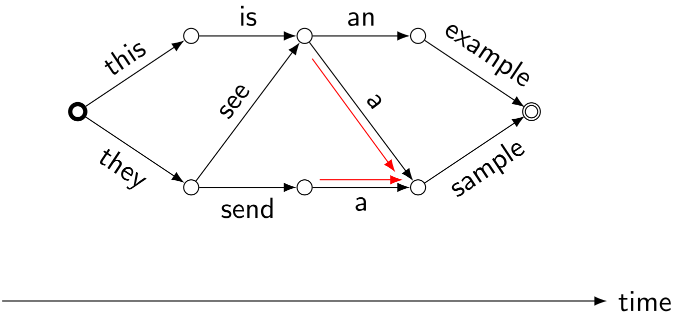
Lattice



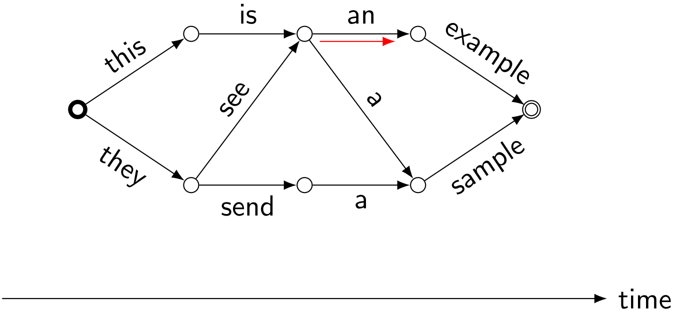
Lattice



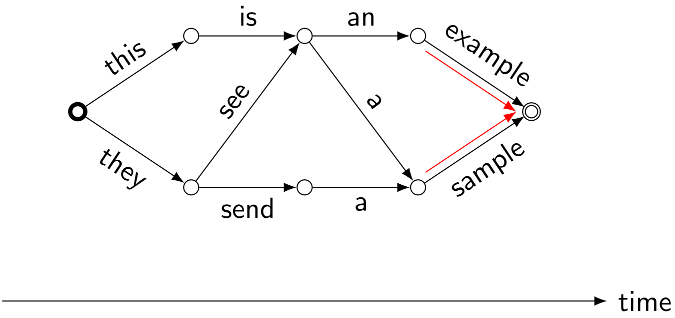
Lattice



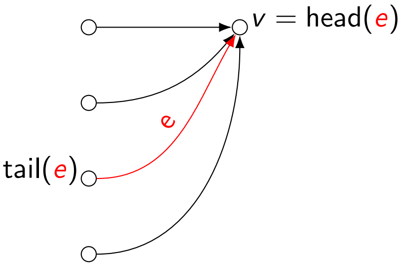
Lattice



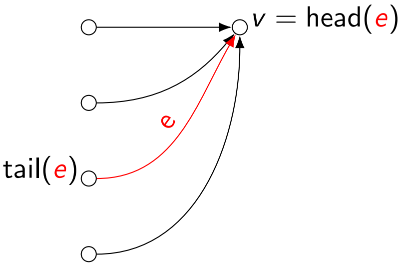
Lattice



Forward-Backward on Graphs



Forward-Backward on Graphs



$$\alpha(v) = \sum_{e \in \text{in}(v)} p(X_e | W_e) \alpha(\text{tail}(e))$$

Forward-Backward on Graphs

- Running the forward algorithm on the lattice only gives

$$\alpha(\text{final}) = \sum_{w' \in D} p(X|W')$$

- Running the forward algorithm on the lattice composed with an LM gives

$$\alpha(\text{final}) = \sum_{w' \in D} p(X|W')p(W')$$

Optimizing MMI

$$p(W|X) = \frac{p(X|W)p(W)}{\sum_{W'} p(X|W')p(W')}$$

- Generate lattice (through beam search)
- Run the forward algorithm
- Compute the gradient
- Do gradient update

Gradient w.r.t. to An Edge

$$\frac{\partial L}{\partial p(X_e|W_e)}$$

Gradient w.r.t. to An Edge

$$\frac{\partial L}{\partial p(X_e|W_e)} = \sum_v \frac{\partial L}{\partial \alpha(v)} \frac{\partial \alpha(v)}{\partial p(X_e|W_e)}$$

Gradient w.r.t. to An Edge

$$\frac{\partial L}{\partial p(X_e|W_e)} = \sum_v \frac{\partial L}{\partial \alpha(v)} \frac{\partial \alpha(v)}{\partial p(X_e|W_e)}$$
$$= \sum_v \frac{\partial L}{\partial \alpha(v)} \alpha(\text{tail}(e)) \mathbb{1}_{v=\text{head}(e)}$$

Gradient w.r.t. to An Edge

$$\frac{\partial L}{\partial p(X_e|W_e)} = \sum_v \frac{\partial L}{\partial \alpha(v)} \frac{\partial \alpha(v)}{\partial p(X_e|W_e)}$$

$$= \sum_v \frac{\partial L}{\partial \alpha(v)} \alpha(\text{tail}(e)) \mathbb{1}_{v=\text{head}(e)}$$

$$= \frac{\partial L}{\partial \alpha(\text{head}(e))} \alpha(\text{tail}(e))$$

Gradient w.r.t. to An Edge

$$\frac{\partial L}{\partial p(X_e|W_e)} = \sum_v \frac{\partial L}{\partial \alpha(v)} \frac{\partial \alpha(v)}{\partial p(X_e|W_e)}$$
$$= \sum_v \frac{\partial L}{\partial \alpha(v)} \alpha(\text{tail}(e)) \mathbb{1}_{v=\text{head}(e)}$$
$$= \frac{\partial L}{\partial \alpha(\text{head}(e))} \alpha(\text{tail}(e))$$

$$\frac{\partial L}{\partial \alpha(u)} = \sum_v \frac{\partial L}{\partial \alpha(v)} \frac{\partial \alpha(v)}{\partial \alpha(u)}$$

Gradient w.r.t. to An Edge

$$
\begin{aligned}
\frac{\partial L}{\partial p(X_e|W_e)} &= \sum_v \frac{\partial L}{\partial \alpha(v)} \frac{\partial \alpha(v)}{\partial p(X_e|W_e)} \\
&= \sum_v \frac{\partial L}{\partial \alpha(v)} \alpha(\text{tail}(e)) \mathbb{1}_{v=\text{head}(e)} \\
&= \frac{\partial L}{\partial \alpha(\text{head}(e))} \alpha(\text{tail}(e))
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial L}{\partial \alpha(u)} &= \sum_v \frac{\partial L}{\partial \alpha(v)} \frac{\partial \alpha(v)}{\partial \alpha(u)} \\
&= \sum_v \frac{\partial L}{\partial \alpha(v)} P(X_e|W_e) \mathbb{1}_{u=\text{tail}(e), v=\text{head}(e)}
\end{aligned}
$$

Gradient w.r.t. to An Edge

$$\frac{\partial L}{\partial p(X_e|W_e)} = \sum_v \frac{\partial L}{\partial \alpha(v)} \frac{\partial \alpha(v)}{\partial p(X_e|W_e)}$$

$$= \sum_v \frac{\partial L}{\partial \alpha(v)} \alpha(\text{tail}(e)) \mathbb{1}_{v=\text{head}(e)}$$

$$= \frac{\partial L}{\partial \alpha(\text{head}(e))} \alpha(\text{tail}(e))$$

$$\frac{\partial L}{\partial \alpha(u)} = \sum_v \frac{\partial L}{\partial \alpha(v)} \frac{\partial \alpha(v)}{\partial \alpha(u)}$$

$$= \sum_v \frac{\partial L}{\partial \alpha(v)} P(X_e|W_e) \mathbb{1}_{u=\text{tail}(e), v=\text{head}(e)}$$

$$= \sum_{e \in \text{out}(u)} \frac{\partial L}{\partial \alpha(\text{head}(e))} P(X_e|W_e)$$

Why the Name MMI?

$$\underset{\lambda}{\operatorname{argmax}} \sum_{X,W} p(X, W) \log \frac{p(X, W)}{P(X)P(W)}$$

Why the Name MMI?

$$\operatorname*{argmax}_{\lambda} \sum_{X,W} p(X,W) \log \frac{p(X,W)}{P(X)P(W)}$$

$$= \operatorname*{argmax}_{\lambda} \sum_{X,W} p(X,W) \log p(W|X) - \sum_{W,X} p(X,W) \log p(W)$$

Why the Name MMI?

$$\operatorname*{argmax}_{\lambda} \sum_{X,W} p(X, W) \log \frac{p(X, W)}{P(X)P(W)}$$

$$= \operatorname*{argmax}_{\lambda} \sum_{X,W} p(X, W) \log p(W|X) - \sum_{W,X} p(X, W) \log p(W)$$

$$= \operatorname*{argmax}_{\lambda} \sum_{X,W} p(X, W) \log p(W|X) - \sum_{W} p(W) \log p(W)$$

Why the Name MMI?

$$
\begin{aligned}
&\operatorname*{argmax}_{\lambda} \sum_{X,W} p(X, W) \log \frac{p(X, W)}{P(X)P(W)} \\
&= \operatorname*{argmax}_{\lambda} \sum_{X,W} p(X, W) \log p(W|X) - \sum_{W,X} p(X, W) \log p(W) \\
&= \operatorname*{argmax}_{\lambda} \sum_{X,W} p(X, W) \log p(W|X) - \sum_{W} p(W) \log p(W) \\
&\approx \operatorname*{argmax}_{\lambda} \frac{1}{N} \sum_{n=1}^{N} \log p(W_n|X_n)
\end{aligned}
$$

Lattice-Free MMI (Povey *et al.*, 2016)

- It is actually possible (just computationally expensive) to compute the denominator $\sum_{W'} p(X|W')p(W')$ exactly with the help of GPU.
- The trick is to realize that the forward algorithm is a matrix multiplication.

Properties of MMI

- It is a discriminative approach.
- It considers a language model.
- It provides the same solution as minimizing the zero-one loss.

Properties of MMI

- It is a discriminative approach.
- It considers a language model.
- It provides the same solution as minimizing the zero-one loss.

$$\mathbb{E}_{W' \sim p(W'|X)}[\mathbb{1}_{W \neq W'}]$$

Properties of MMI

- It is a discriminative approach.
- It considers a language model.
- It provides the same solution as minimizing the zero-one loss.

$$\mathbb{E}_{W' \sim p(W'|X)}[\mathbb{1}_{W \neq W'}] = 1 - \mathbb{E}_{W' \sim p(W'|X)}[\mathbb{1}_{W = W'}]$$

Properties of MMI

- It is a discriminative approach.
- It considers a language model.
- It provides the same solution as minimizing the zero-one loss.

$$\begin{aligned}\mathbb{E}_{W'\sim p(W'|X)}[\mathbb{1}_{W\neq W'}] &= 1 - \mathbb{E}_{W'\sim p(W'|X)}[\mathbb{1}_{W=W'}] \\ &= 1 - p(W|X)\end{aligned}$$

Properties of MMI

- It is a discriminative approach.
- It considers a language model.
- It provides the same solution as minimizing the zero-one loss.

$$\begin{aligned}\mathbb{E}_{W' \sim p(W'|X)}[\mathbb{1}_{W \neq W'}] &= 1 - \mathbb{E}_{W' \sim p(W'|X)}[\mathbb{1}_{W = W'}] \\ &= 1 - p(W|X)\end{aligned}$$

$$\operatorname*{argmin}_{\lambda} \mathbb{E}_{W' \sim p(W'|X)}[\mathbb{1}_{W \neq W'}] = \operatorname*{argmax}_{\lambda} p(W|X)$$

Minimum Bayes Risk (MBR)

$$\operatorname*{argmax}_{\lambda} \mathbb{E}_{W' \sim p(W'|X)}[\text{cost}(W, W')]$$

Minimum Bayes Risk (MBR)

$$\operatorname*{argmax}_{\lambda} \mathbb{E}_{W' \sim p(W'|X)}[\text{cost}(W, W')]$$

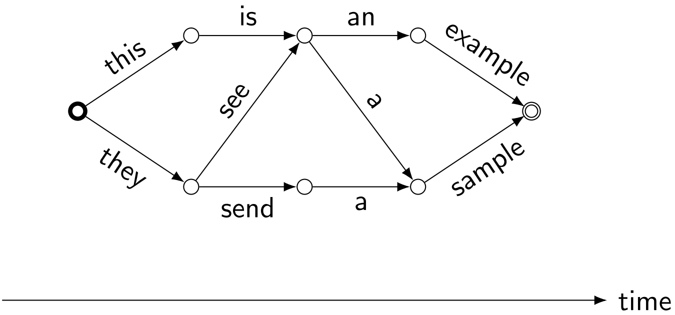
- Allows partial credit
- Allows a user-defined cost function

Minimum Bayes Risk (MBR)

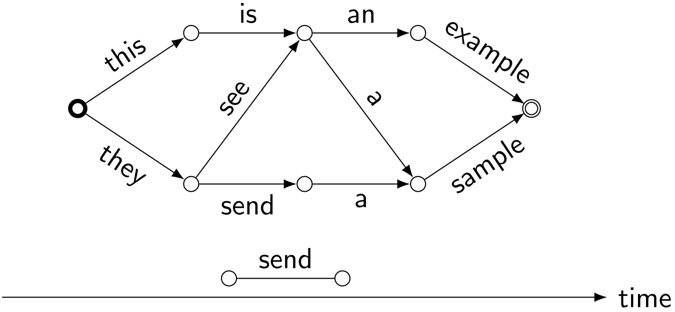
$$\mathbb{E}_{W' \sim p(W'|X)}[\text{cost}(W, W')] = \sum_{W'} p(W'|X)\text{cost}(W, W')$$
$$= \frac{\sum_{W'} p(X|W')p(W')\text{cost}(W, W')}{\sum_{W''} p(X|W'')p(W'')}$$

- Both numerators and denominators require a lattice.
- The cost function needs to decompose according to a lattice, i.e., each edge having a cost.
- WER(W, W') does not decompose according to a lattice.

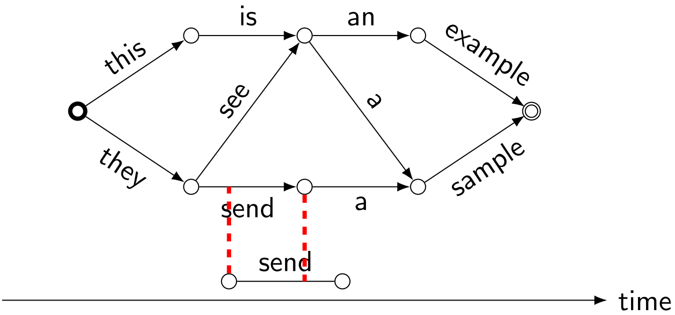
Minimum Bayes Risk (MBR)



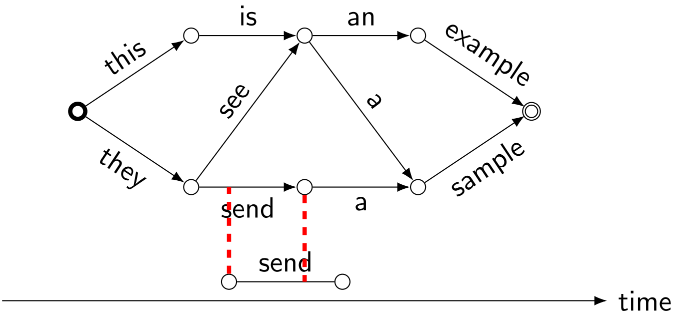
Minimum Bayes Risk (MBR)



Minimum Bayes Risk (MBR)



Minimum Bayes Risk (MBR)



- If the cost is at the phone level, the objective is called Minimum Phone Error (MPE) (Povey and Woodland, 2002).
- If the cost is at the word level, the objective is called Minimum Word Error (MWE) (Povey and Woodland, 2002).

Summary

- Discriminative vs Generative Training
- Maximum Mutual Information
- Forward-Backward on Graphs
- Minimum Bayes Risk