Conditional Random Fields and Direct Decoding for Speech and Language Processing

Eric Fosler-Lussier
The Ohio State University
Geoff Zweig
Microsoft

What we will cover

- Tutorial introduces basics of direct probabilistic models
  - What is a direct model, and how does it relate to speech and language processing?
  - How do I train a direct model?
  - How have direct models been used in speech and language processing?

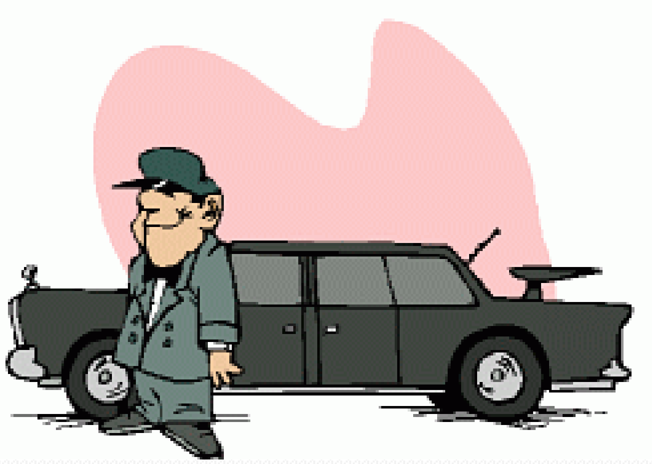
Overview

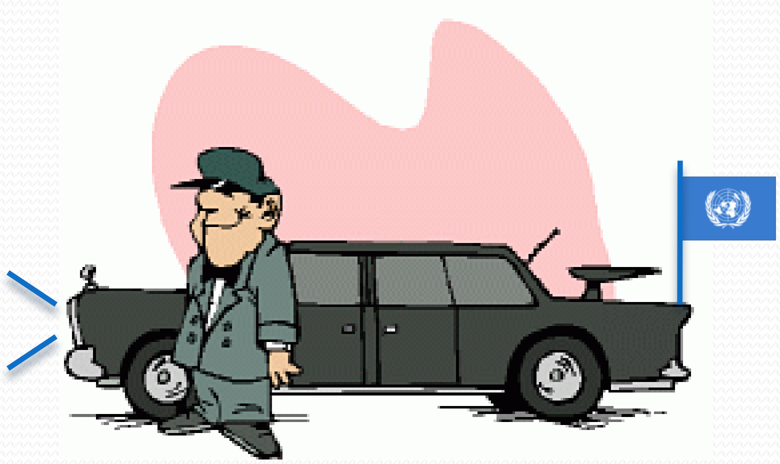
- Part 1: Background and Taxonomy
  - Generative vs. Direct models
  - Descriptions of models for classification, sequence recognition (observed and hidden)
- Break
- Part 2: Algorithms & Case Studies
  - Training/decoding algorithms
  - CRF study using phonological features for ASR
  - Segmental CRF study for ASR
  - NLP case studies (if time)

Part 1:
Background and Taxonomy

A first thought experiment

- You're observing a limousine – is a diplomat inside?
  - Can observe:
    - Whether the car has flashing lights
    - Whether the car has flags





The Diplomat problem

- We have observed Boolean variables: lights and flag
- We want to predict if car contains a diplomat

$$P(Diplomat \mid Lights, Flag)$$

A generative approach: Naïve Bayes

- Generative approaches model observations as being *generated* by the underlying class – we observe:
  - Limos carrying diplomats have flags 50% of the time
  - Limos carrying diplomats have flashing lights 70%
  - Limos not carrying diplomats: flags 5%, lights 30%
- NB: Compute posterior by Bayes' rule

$$P(Diplomat \mid Lights, Flag) = \frac{P(Lights, Flag \mid Diplomat) P(Diplomat)}{P(Lights, Flag)}$$

A generative approach: Naïve Bayes

- Generative approaches model observations as being *generated* by the underlying class – we observe:
  - Limos carrying diplomats have flags 50% of the time
  - Limos carrying diplomats have flashing lights 70%
  - Limos not carrying diplomats: flags 5%, lights 30%
- NB: Compute posterior by Bayes' rule
  - ...and then assume conditional independence

$$P(Dmat \mid Lights, Flag) = \frac{P(Lights \mid Dmat) P(Flag \mid Dmat) P(Dmat)}{P(Lights, Flag)}$$

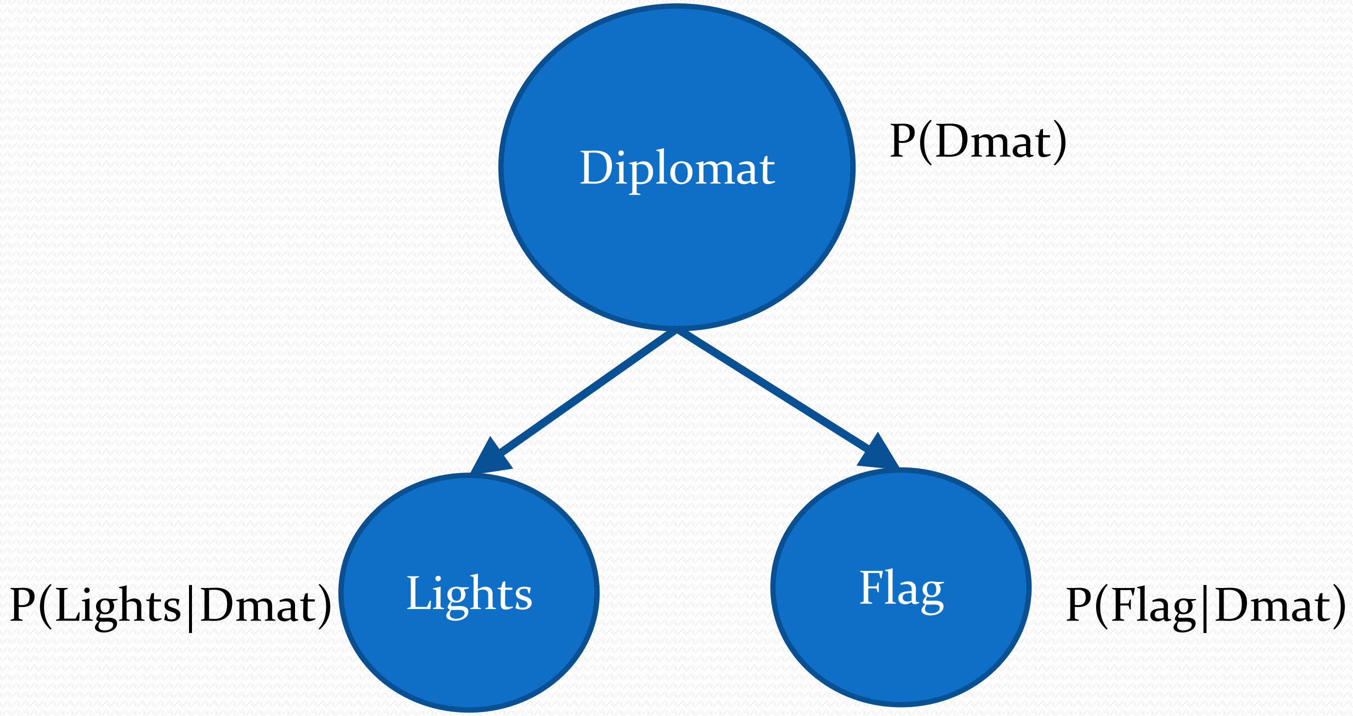
A generative approach: Naïve Bayes

- NB: Compute posterior by Bayes' rule
  - …and then assume conditional independence
  - P(Lights, Flag) is a normalizing term
    - Can replace this with normalization constant Z

$$P(Dmat \mid Lights, Flag) = \frac{P(Lights \mid Dmat) P(Flag \mid Dmat) P(Dmat)}{Z}$$

Graphical model for Naïve Bayes

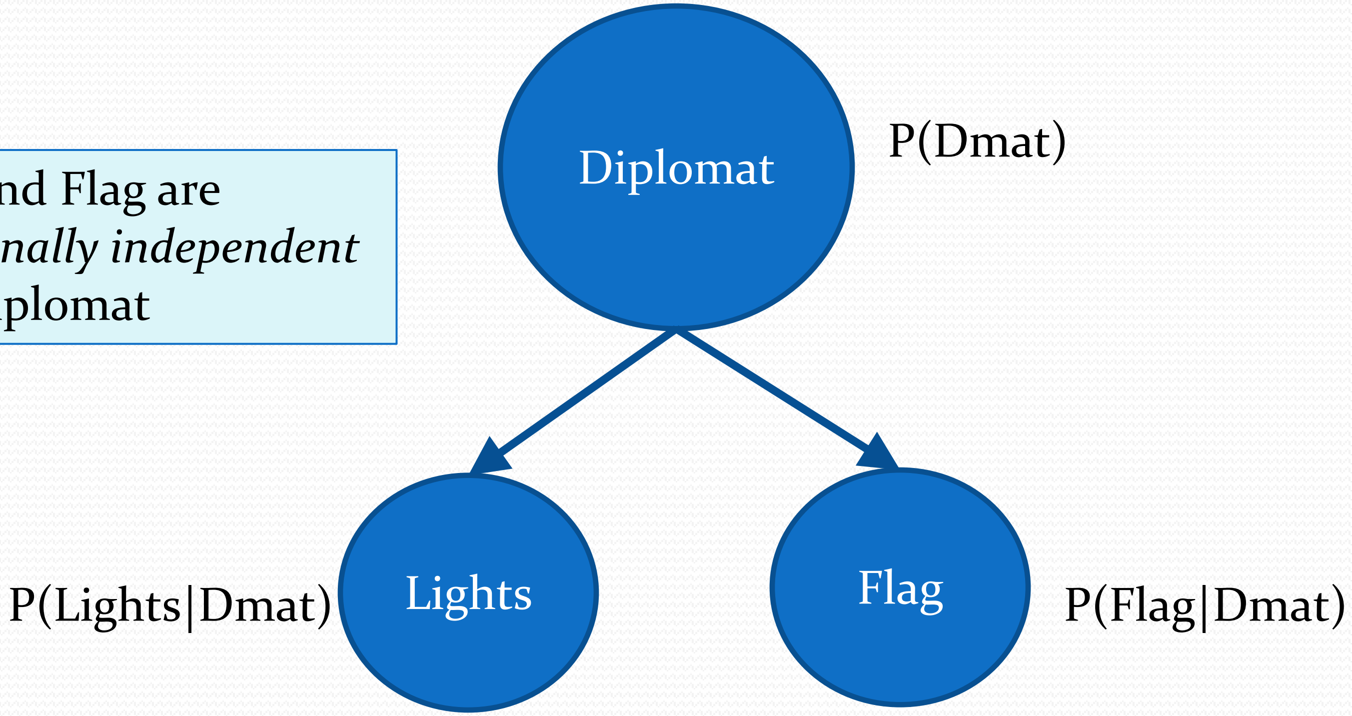
$$P(Dmat \mid Lights, Flag) = \frac{P(Lights \mid Dmat) P(Flag \mid Dmat) P(Dmat)}{Z}$$



Graphical model for Naïve Bayes

$$P(Dmat \mid Lights, Flag) = \frac{P(Lights \mid Dmat)P(Flag \mid Dmat)P(Dmat)}{Z}$$

Lights and Flag are *conditionally independent* given Diplomat



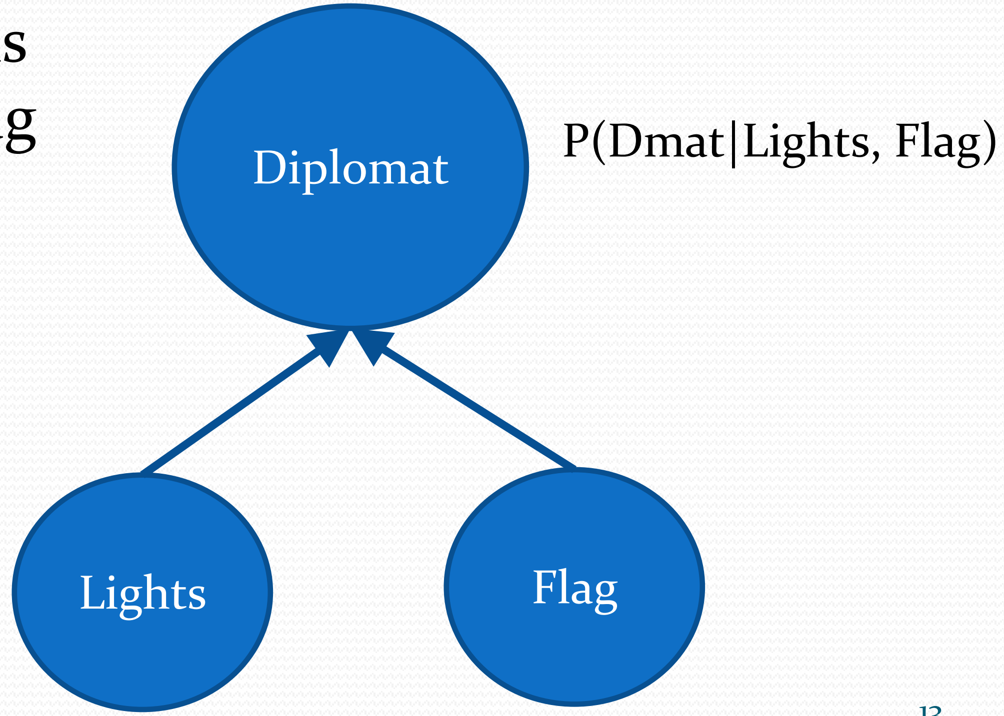
Correlated evidence in Naïve Bayes

- Conditional independence says "given a value of Diplomat, Lights and Flag are independent"
- Consider the case where lights are *always* flashing when the car has flags
  - Evidence gets double counted; NB is *overconfident*
  - May not be a problem in practice – problem dependent
  - (HMMs have similar assumptions: observations are independent given HMM state sequence.)

$$P(Dmat \mid Lights, Flag) = \frac{P(Lights \mid Dmat)P(Flag \mid Dmat)P(Dmat)}{Z}$$

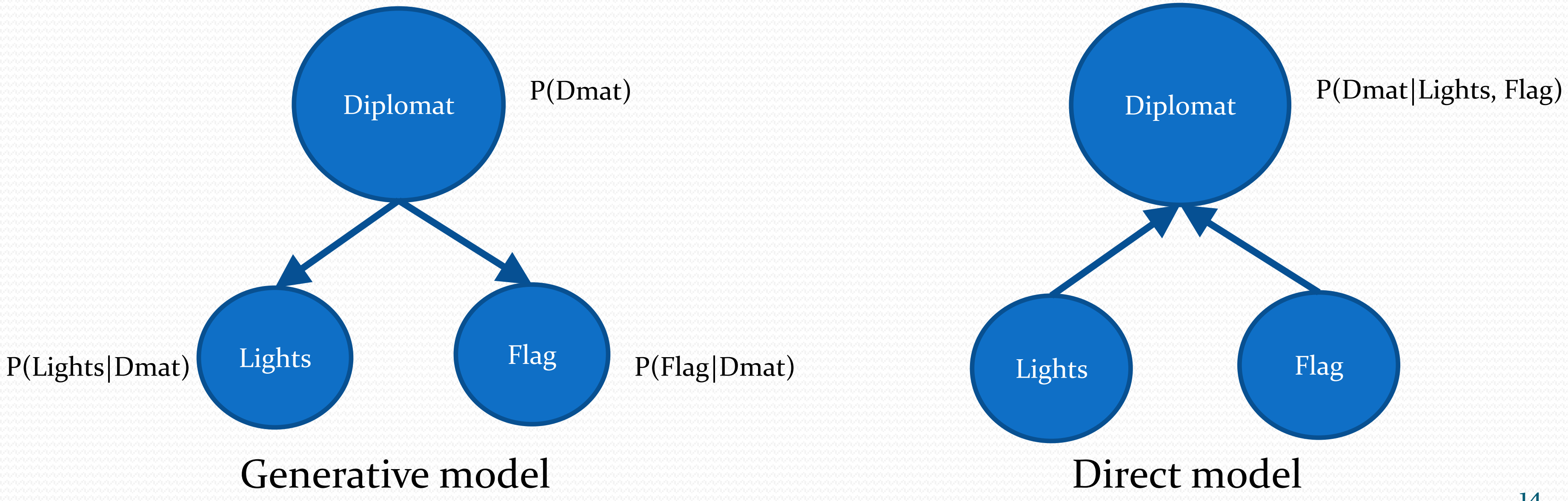
Reversing the arrows: Direct modeling

- P(Diplomat|Lights,Flag) can be directly modeled
  - We compute a probability distribution directly without Bayes' rule
  - Can handle interactions between Lights and Flag evidence
- P(Lights) and P(Flag) do *not* need to be modeled



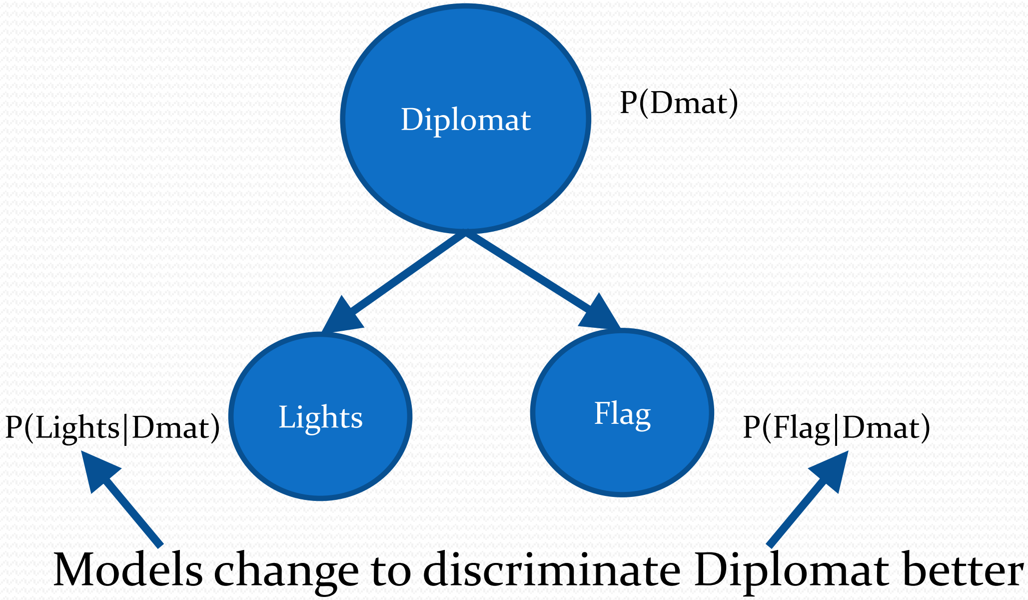
Direct vs. Discriminative

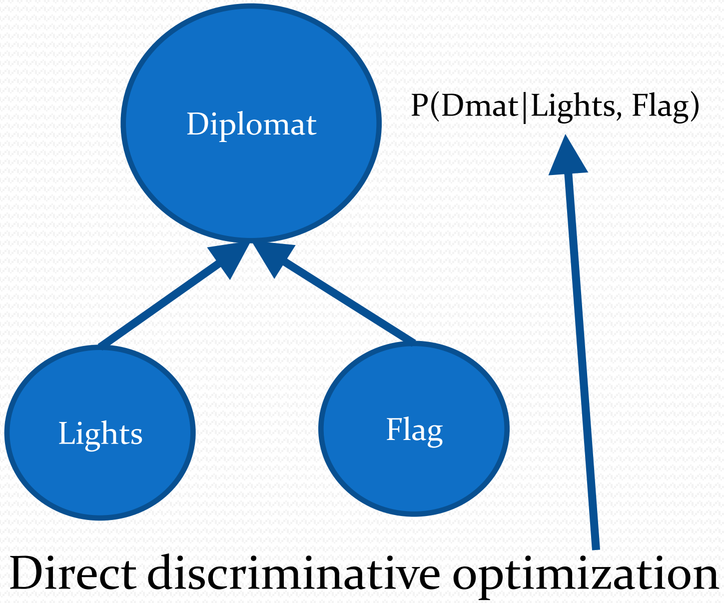
- Isn't this just discriminative training? (No.)
  - Direct **model**: directly predict posterior of hidden variable
  - Discriminative **training**: adjust model parameters to {separate classes, improve posterior, minimize classification error,…}



Direct vs. Discriminative

- Generative models can be trained discriminatively
- Direct models inherently try to discriminate between classes



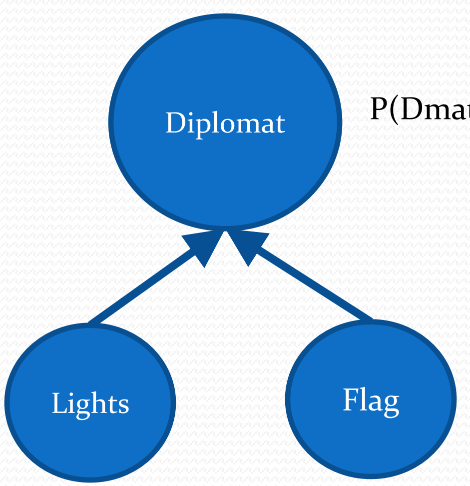


Pros and cons of direct modeling

- Pro:
  - Often can allow modeling of interacting data features
  - Can require fewer parameters because there is no observation model
    - Observations are usually treated as *fixed* and don't require a probabilistic model
- Con:
  - Typically slower to train
    - Most training criteria have no closed-form solutions

A simple direct model: Maximum Entropy

- Our direct example didn't have a particular form for the probability P(Dmat|Lights, Flag)
- A maximum entropy model uses a log-linear combination of weighted features in probability model



P(Dmat|Lights, Flag)

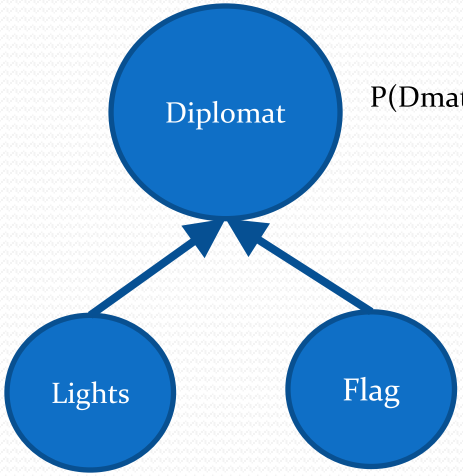
feature of the data for class j

learned weight

$$P(Dmat = j \mid Lights, Flag) = \frac{\exp(\sum_i \lambda_{i,j} f_{i,j})}{\sum_{j'} \exp(\sum_i \lambda_{i,j'} f_{i,j'})}$$

A simple direct model: Maximum Entropy

- Denominator of the equation is again normalization term (replace with Z)
- Question: what are $f_{i,j}$ and how does this correspond to our problem?



P(Dmat|Lights, Flag)

feature of the data for class j

learned weight

$$P(Dmat = j \mid Lights, Flag) = \frac{\exp(\sum_i \lambda_{i,j} f_{i,j})}{Z}$$

Diplomat Maximum Entropy

- Here are two features ($f_{i,j}$) that we can use:
  - $f_{0,True}$=1 if car has a diplomat and has a flag
  - $f_{1,False}$=1 if car has no diplomat but has flashing lights
    - (Could have complementary features as well but left out for simplification.)
- Example dataset with the following statistics
    - Diplomats occur in 50% of cars in dataset
    - P(Flag=true|Diplomat=true) = 0.9 in dataset
    - P(Flag=true|Diplomat=false) = 0.2 in dataset
    - P(Lights=true|Diplomat=false) = 0.7 in dataset
    - P(Lights=true|Diplomat=true) = 0.5 in dataset

Diplomat Maximum Entropy

- The MaxEnt formulation using these two features is:

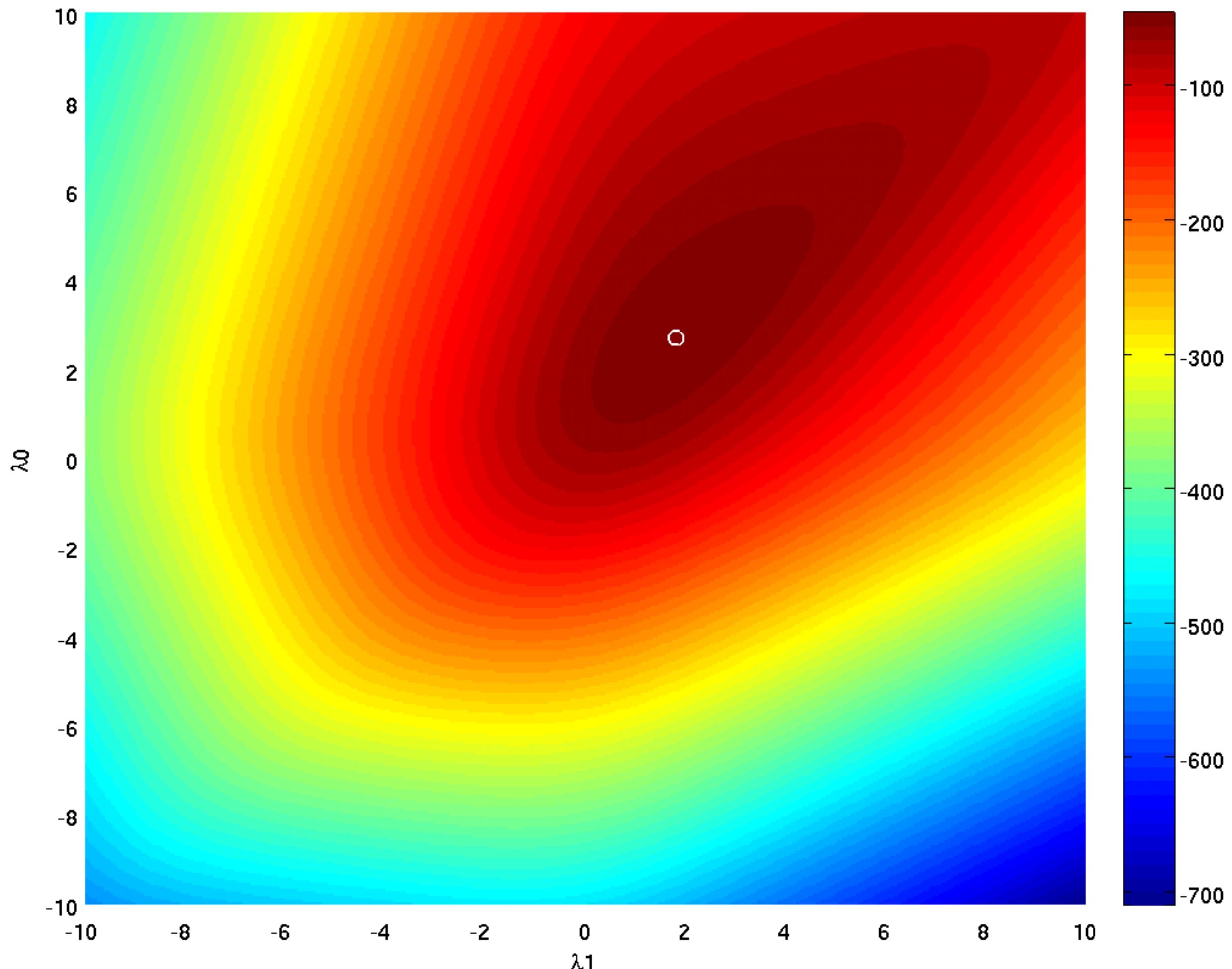
$$P(Dmat = true \mid Flag, Light) = \exp(\lambda_{true} + \lambda_{0,T} f_{0,T})/Z$$
$$P(Dmat = false \mid Flag, Light) = \exp(\lambda_{false} + \lambda_{1,F} f_{1,F})/Z$$

where λ_{true} and λ_{false} are bias terms to adjust for frequency of labels.

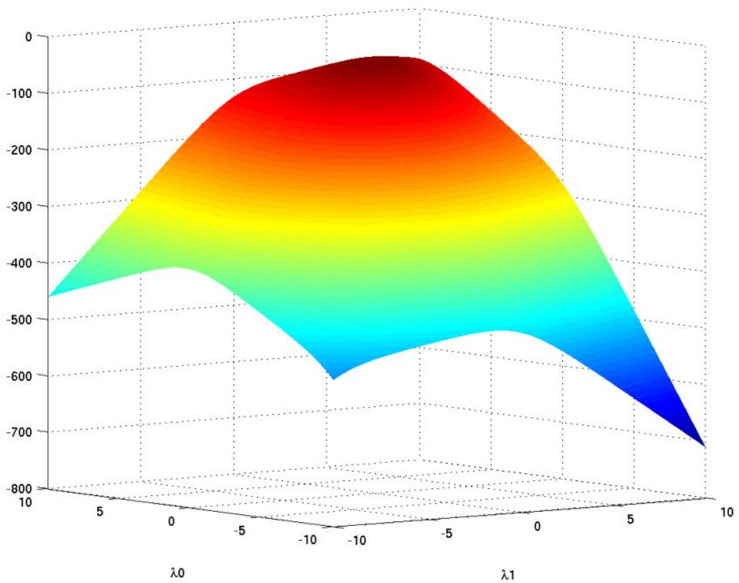
- Fix the bias terms to both be 1. What happens to probability of Diplomat on dataset as other lambdas vary?

$f_{0,T}$=1 if car has a diplomat and has a flag
$f_{1,F}$=1 if car has no diplomat but has flashing lights

Log probability of Diplomat over dataset as MaxEnt lambdas vary



Finding optimal lambdas



Same picture in 3-d:
Conditional probability of dataset

- Good news: conditional probability of dataset is convex for MaxEnt
- Bad news: as number of features grows, finding maximum in so many dimensions can be slooow.
  - Various gradient search or optimization techniques can be used (coming later).

MaxEnt-style models in practice

- Several examples of MaxEnt models in speech & language processing
  - Whole-sentence language models (Rosenfeld, Chen & Zhu, 2001)
    - Predict probability of whole sentence given features over correlated features (word n-grams, class n-grams, …)
    - Good for rescoring hypotheses in speech, MT, etc…
  - Multi-layer perceptrons
    - MLP can really be thought of as MaxEnt models with automatically learned feature functions
      - MLP gives local posterior classification of frame
      - Sequence recognition through Hybrid or Tandem MLP-HMM
    - Softmax-trained Single Layer Perceptron == MaxEnt model

MaxEnt-style models in practice

- Several examples of MaxEnt models in speech & language processing
  - Flat Direct Models for ASR (Heigold et al. 2009)
    - Choose complete hypothesis from list (rather than a sequence of words)
      - Doesn't have to match exact words (auto rental=rent-a-car)
    - Good for large-scale list choice tasks, e.g. voice search
    - What do features look like?

Flat Direct Model Features: Decomposable features

- Decompose features $F(W,X) = \Phi(W)\Psi(X)$
- $\Phi(W)$ is a feature of the words
  - e.g. "The last word ends in s"
  - "The word Restaurant is present"
- $\Psi(X)$ is a feature of the acoustics
  - e.g. "The distance to the *Restaurant* template is greater than 100"
  - "The HMM for *Washington* is among the 10 likeliest"
- $\Phi(W)\Psi(X)$ is the conjunction; measures consistency
  - e.g. "The hypothesis ends is s" and my "s-at-the-end" acoustic detector has fired

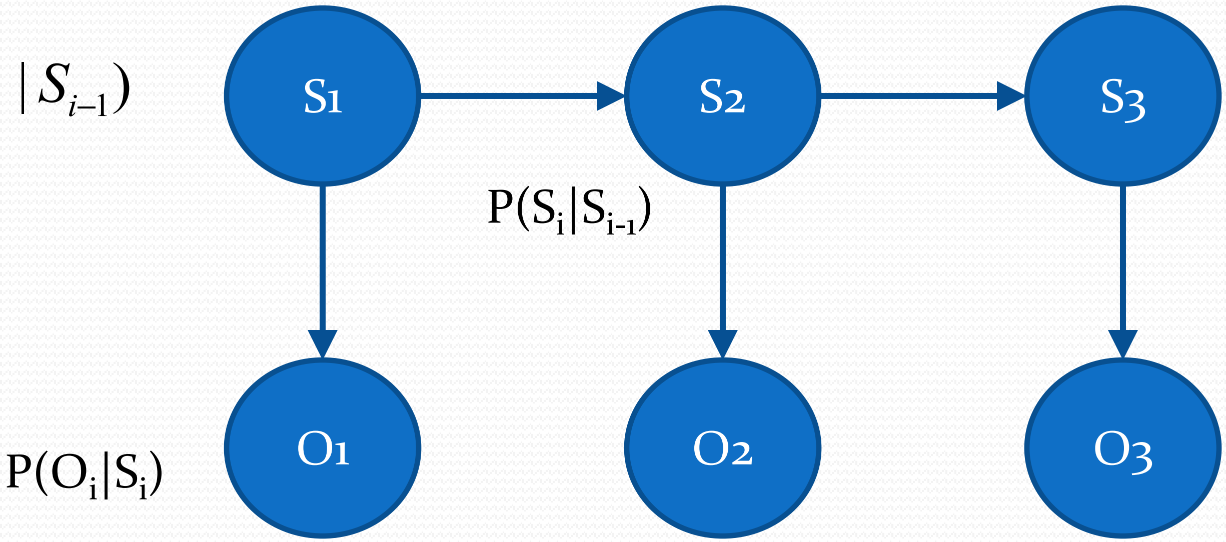
Generalization

- People normally think of Maximum Entropy for classification among a predefined set
- But $F(W,X) = \Phi(W)\Psi(X)$ essentially measures consistency between W and X
- These features are defined for arbitrary W.
- For example, "*Restaurants* is present and my *s-at-the-end* detector has fired" can be true for either "Mexican Restaurants or Italian Restaurants"

Direct sequence modeling

- In speech and language processing, usually want to operate over sequences, not single classifications
- Consider a common generative sequence model – the Hidden Markov Model – relating states (S) to obs. (O)

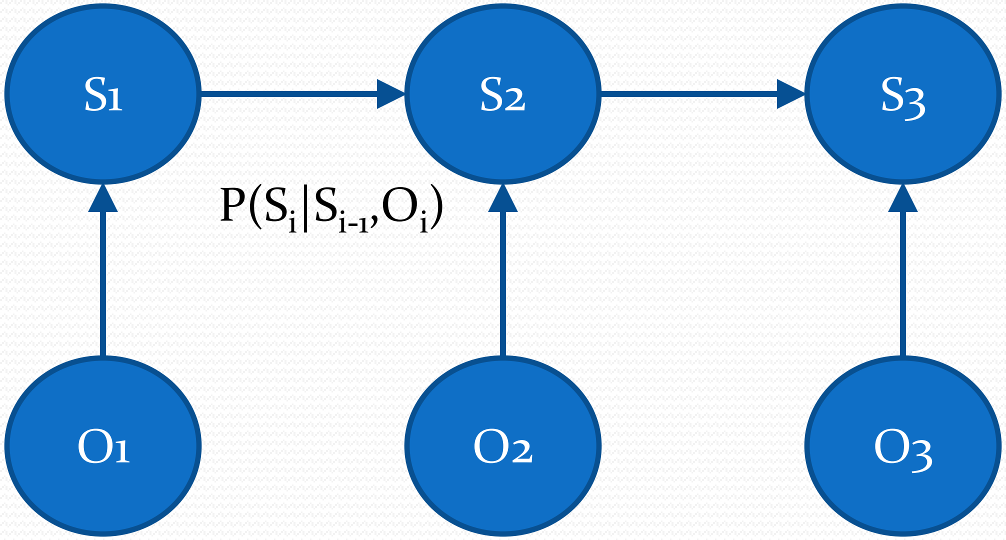
$$P(S,O) = \prod_i P(O_i \mid S_i) P(S_i \mid S_{i-1})$$



Direct sequence modeling

- In speech and language processing, usually want to operate over sequences, not single classifications
- What happens if we "change the direction" of arrows of an HMM? A direct model of P(S|O).

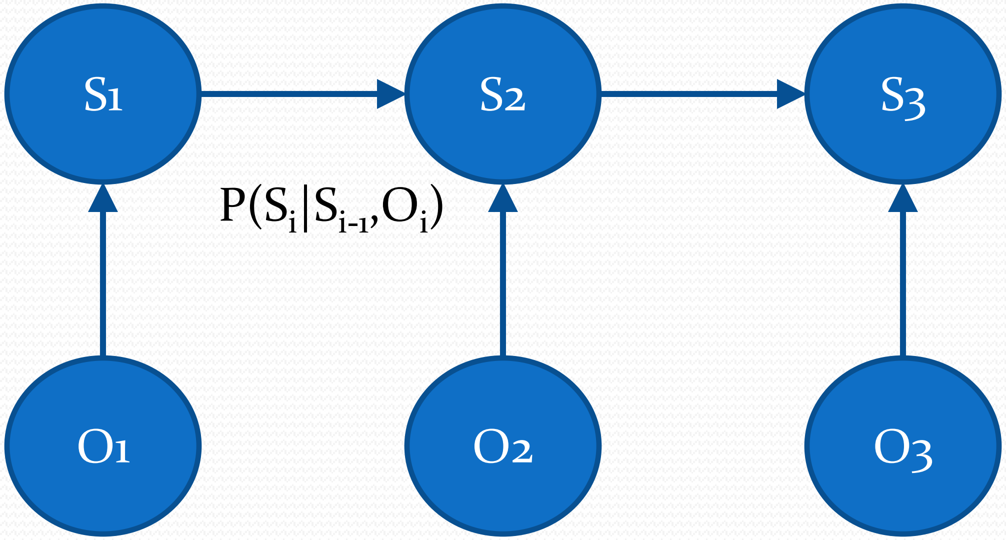
$$P(S \mid O) = P(S_1 \mid O_1) \prod_{i>1} P(S_i \mid S_{i-1}, O_i)$$



MEMMs

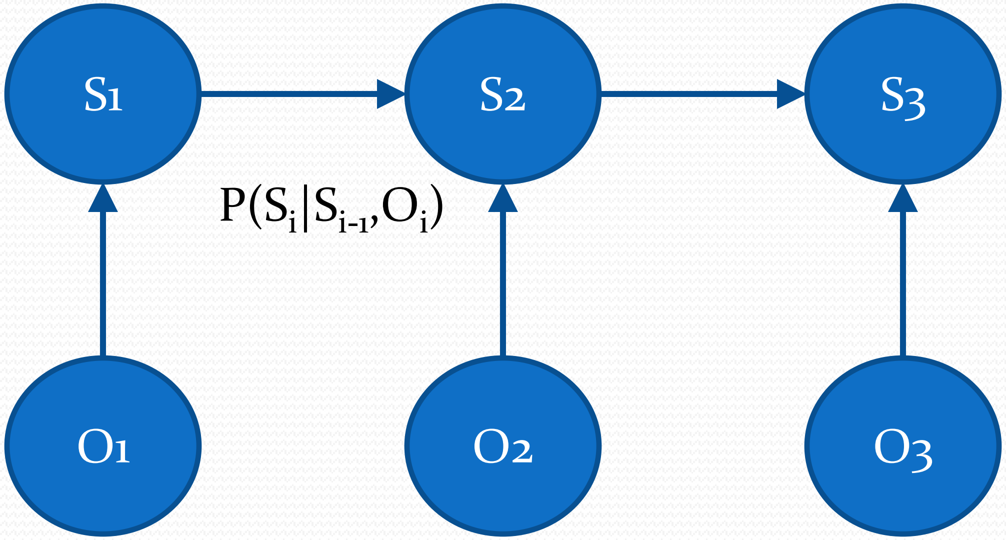
- If a log linear term is used for $P(S_i|S_{i-1},O_i)$ then this is a Maximum Entropy Markov Model (MEMM)
(Ratnaparkhi 1996, McCallum, Freitag & Pereira 2000)
  - Like MaxEnt, we take features of the observations and learn a weighted model

$$P(S \mid O) = P(S_1 \mid O_1) \prod_{i>1} P(S_i \mid S_{i-1}, O_i)$$
$$= \exp\left(\sum_j \sum_i \lambda_j f_j(S_{i-1}, S_i, O, i) \right)$$



MEMMs

- Unlike HMMs, transitions between states can now depend on acoustics in MEMMs
  - However, unlike HMM, MEMMs can ignore observations
    - If $P(S_i=x|S_{i-1}=y)=1$, then $P(S_i=x|S_{i-1}=y,O_i)=1$ for all O_i *(label bias)*
    - Problem in practice?



MEMMs in language processing

- One prominent example in part-of-speech tagging is the Ratnaparkhi “MaxEnt” tagger (1996)
  - Produce POS tags based on word history features
  - Really an MEMM because it includes the previously assigned tags as part of its history
- Kuo and Gao (2003-6) developed “Maximum Entropy Direct Models” for ASR
  - Again, an MEMM, this time over speech frames
  - Features: what are the IDs of the closest Gaussians to this point?

Joint sequence models

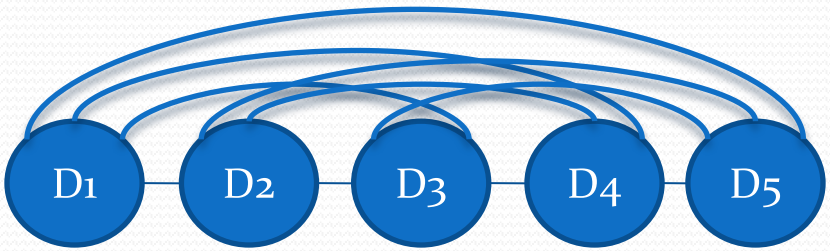
- Label bias problem: previous "decisions" may restrict the influence of future observations
  - Harder for the system to know that it was following a bad path
- Idea: what if we had one big maximum entropy model where we compute the *joint* probability of hidden variables given observations?
  - Many-diplomat problem: $P(Dmat_1 \ldots Dmat_N | Flag_1 \ldots Flag_N, Lights_1 \ldots Lights_N)$
  - Problem: State space is exponential in length
    - Diplomat problem: $O(2^N)$

Factorization of joint sequences

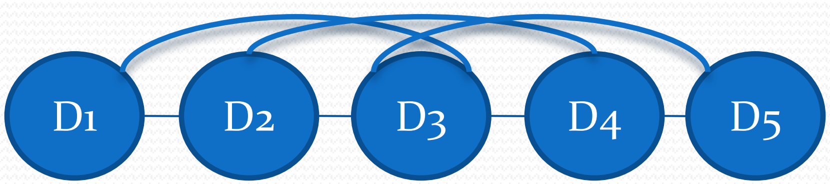
- What we want is a *factorization* that will allow us to decrease the size of the state space
  - Define a Markov graph to describe factorization: *Markov Random Field (MRF)*
  - Neighbors in graph contribute to the probability distribution
    - More formally: probability distribution is factored by the cliques in a graph

Markov Random Fields (MRFs)

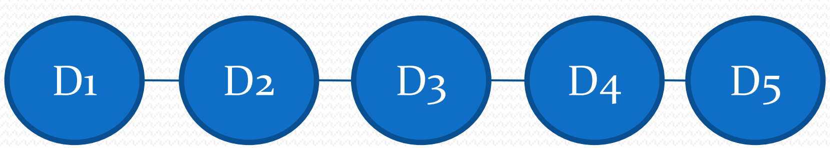
- MRFs are undirected (joint) graphical models
- Cliques define probability distribution
  - Configuration size of each clique is the effective state space
  - Consider 5-diplomat series



One 5-clique (fully connected)
Effective state space is 2^5 (MaxEnt)



Three 3-cliques (1-2-3, 2-3-4, 3-4-5)
Effective state space is 2^3

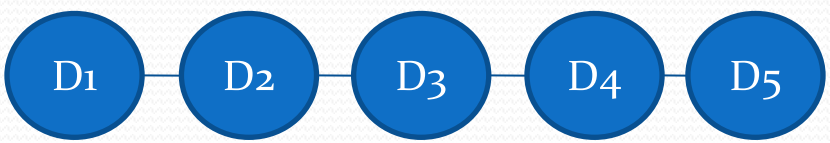


Four 2-cliques (1-2, 2-3, 3-4, 4-5)
Effective state space is 2^2

Hammersley-Clifford Theorem

- Hammersley-Clifford Theorem related MRFs to Gibbs probability distributions
  - If you can express the probability of a graph configuration as a product of potentials on the cliques (Gibbs distribution), then the graph is an MRF

$$P(D) = \prod_{c \in cliques(D)} \phi(c)$$



$$P(D) = \phi(D_1, D_2)\phi(D_2, D_3)\phi(D_3, D_4)\phi(D_4, D_5)$$

  - The potentials, however, must be positive
    - True if $\phi(c)=\exp(\Sigma\ \lambda f(c))$ *(log linear form)*

Conditional Random Fields (CRFs)

- When the MRF is conditioned on observations, this is known as a Conditional Random Field (CRF) (Lafferty, McCallum & Pereira, 2001)
  - Assuming log-linear form (true of almost all CRFs), then probability is determined by weighted functions (f_i) of the clique (c) and the observations (O)

$$P(D \mid O) = \frac{1}{Z} \prod_{c \in cliques(D)} \exp\left(\sum_i \lambda_i f_i(c, O) \right)$$

$$P(D \mid O) = \frac{1}{Z} \exp\left(\sum_{c \in cliques(D)} \sum_i \lambda_i f_i(c, O) \right)$$

$$\log(P(D \mid O)) = \sum_{c \in cliques(D)} \sum_i \lambda_i f_i(c, O) - \log(Z)$$

Conditional Random Fields (CRFs)

- When the MRF is conditioned on observations, this is known as a Conditional Random Field (CRF) (Lafferty, McCallum & Pereira, 2001)
  - Assuming log-linear form (true of almost all CRFs), then probability is determined by weighted functions (f_i) of the clique (c) and the observations (O)

$$P(D \mid O) = \frac{1}{Z} \prod_{c \in cliques(D)} \exp\left(\sum_i \lambda_i f_i(c,O) \right)$$

$$P(D \mid O) = \frac{1}{Z} \exp\left(\sum_{c \in cliques(D)} \sum_i \lambda_i f_i(c,O) \right)$$

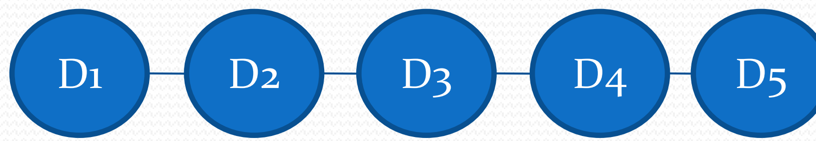
$$\log(P(D \mid O)) = \sum_{c \in cliques(D)} \sum_i \lambda_i f_i(c,O) - \log(Z)$$

For general graphs, computing this quantity is #P-hard, requiring approximate inference.

However, for special graphs the complexity is lower. For example, *linear chain CRFs* have polynomial time algorithms.

Log-linear Linear Chain CRFs

- Linear-chain CRFs have a 1st order Markov backbone
  - Feature templates for a HMM-like CRF structure for the Diplomat problem



- $f_{Bias}(D_i=x, i)$ is 1 iff $D_i=x$
- $f_{Trans}(D_i=x, D_{i+1}=y, i)$ is 1 iff $D_i=x$ and $D_{i+1}=y$
- $f_{Flag}(D_i=x, Flag_i=y, i)$ is 1 iff $D_i=x$ and $Flag_i=y$
- $f_{Lights}(D_i=x, Lights_i=y, i)$ is 1 iff $D_i=x$ and $Lights_i=y$

  - With a bit of subscript liberty, the equation is

$$P(D_1 \ldots D_5 \mid F_{1\ldots5}, L_{1\ldots5}) = \frac{1}{Z(F,L)} \exp\left(\sum_{i=1}^{5} \lambda_B f_{Bias}(D_i) + \sum_{i=1}^{5} \lambda_F f_{Flag}(D_i, F_i) + \sum_{i=1}^{5} \lambda_L f_{Lights}(D_i, L_i) + \sum_{i=1}^{4} \lambda_T f_{Trans}(D_i, D_{i+1}) \right)$$

Log-linear Linear Chain CRFs

- In the previous example, the transitions did not depend on the observations (HMM-like)
  - In general, transitions **may** depend on observations (MEMM-like)

- General form of linear chain CRF groups features as state features (bias, flag, lights) or transition features
  - Let *s* range over state features, *t* over transition features
  - *i* indexes into the sequence to pick out relevant observations

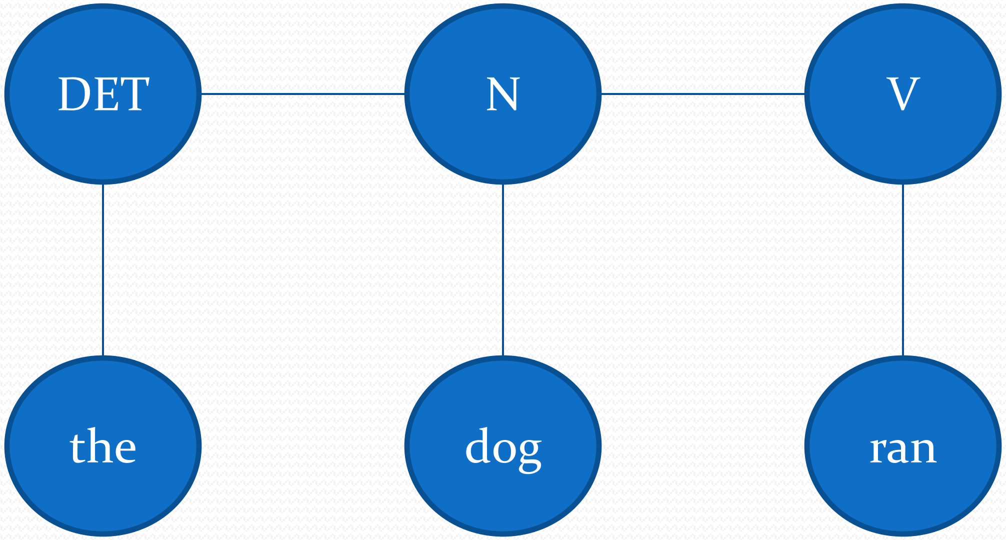
$$P(D \mid O) = \frac{1}{Z(O)} \exp\left(\sum_{s \in stateFtrs} \sum_{i=1}^{n} \lambda_s f_s(D_i, O, i) + \sum_{t \in transFtrs} \sum_{i=1}^{n-1} \lambda_t f_t(D_i, D_{i+1}, O, i) \right)$$

A quick note on features for ASR

- Both MEMMs and CRFS require the definition of feature functions
  - Somewhat obvious in NLP (word id, POS tag, parse structure)
- In ASR, need some sort of "symbolic" representation of the acoustics
  - What are the closest Gaussians (Kuo & Gao, Hifny & Renals)
  - Sufficient statistics (Layton & Gales, Gunawardana et al)
    - With sufficient statistics, can exactly replicate single Gaussian HMM in CRF, or mixture of Gaussians in HCRF (next!)
  - Other classifiers (e.g. MLPs) (Morris & Fosler-Lussier)
  - Phoneme/Multi-Phone detections (Zweig & Nguyen)

Sequencing: Hidden Structure (1)

- So far there has been a 1-to-1 correspondence between labels and observations
  - And it has been fully observed in training



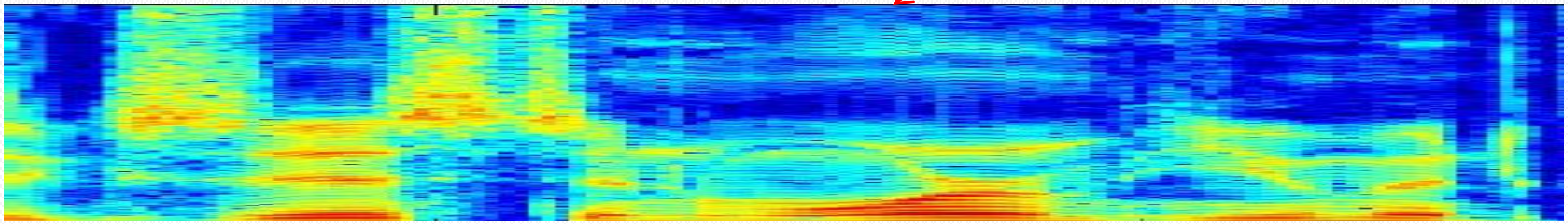
Sequencing: Hidden Structure (2)

- But this is often not the case for speech recognition
- Suppose we have training data like this:

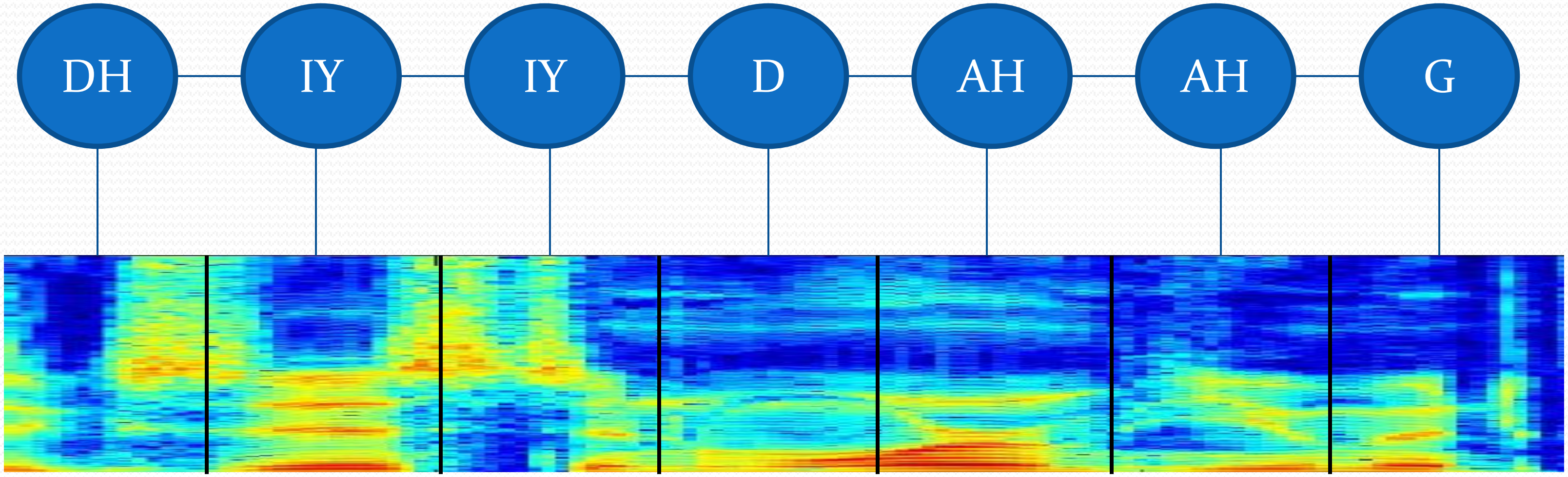
Transcript

“The Dog”

Audio (spectral representation)

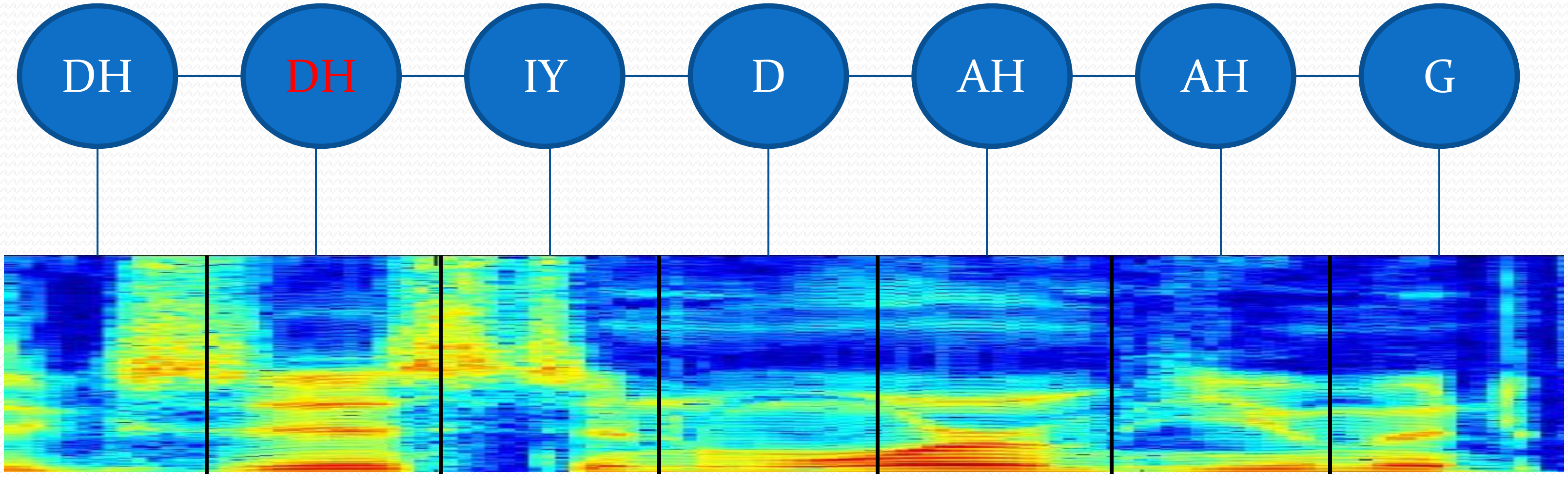


Sequencing: Hidden Structure (3)



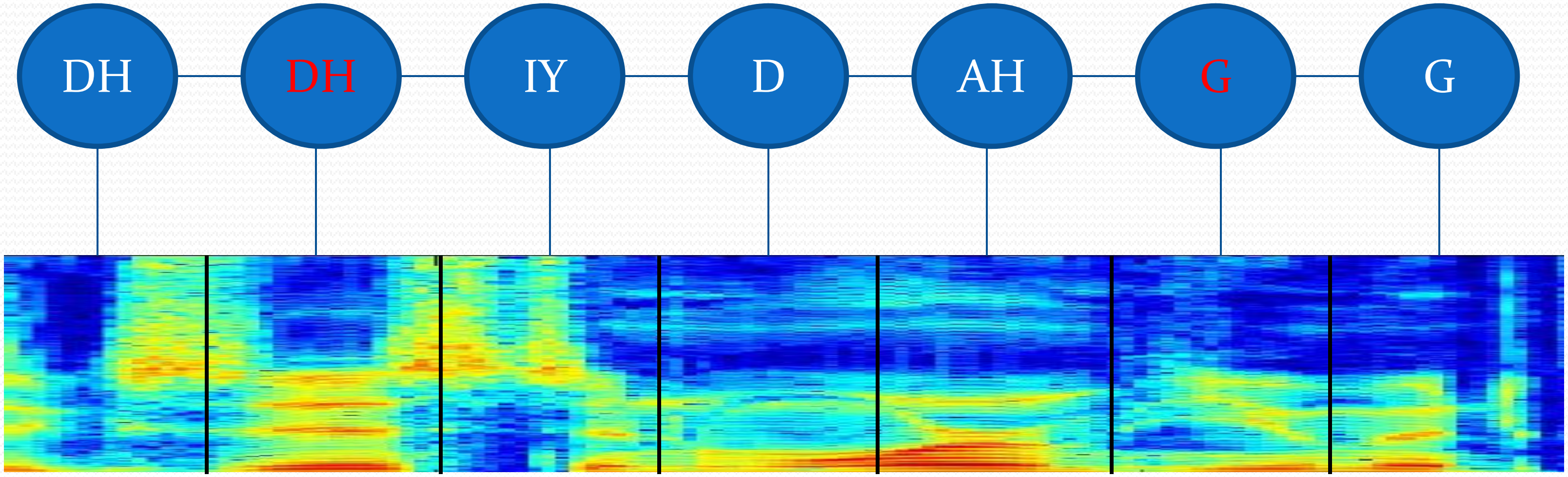
Is “The dog” segmented like this?

Sequencing: Hidden Structure (3)



Or like this?

Sequencing: Hidden Structure (3)



Or maybe like this?

=> An added layer of complexity

This Can Apply in NLP as well

callee caller
Hey John Deb Abrams calling how are you

callee caller
Hey John Deb Abrams calling how are you

How should this be segmented?
Note that a segment level feature indicating that "Deb Abrams" is a 'good' name would be useful

Approaches to Hidden Structure

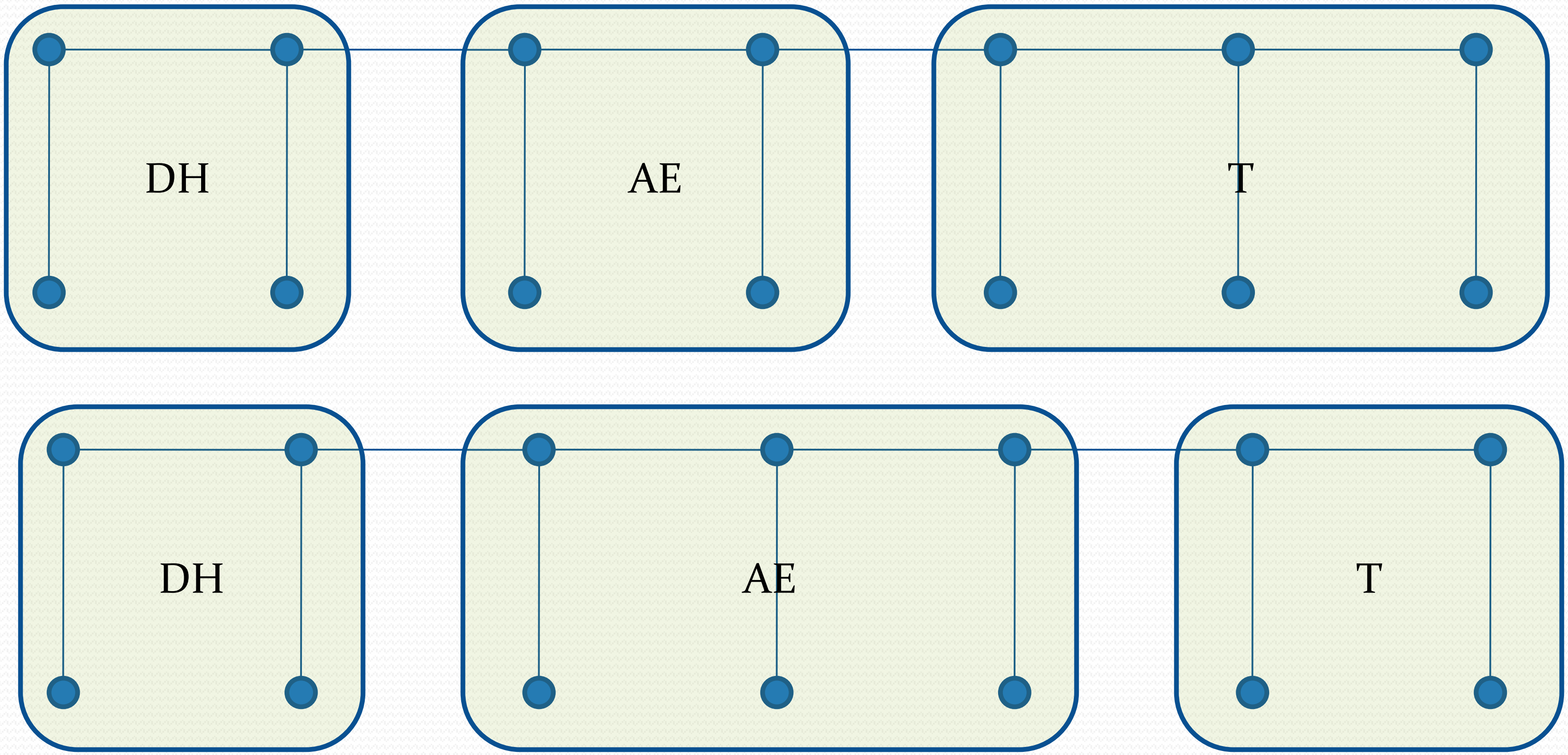
- Hidden CRFs (HRCFs)
  - Gunawardana et al., 2005
- Semi-Markov CRFs
  - Sarawagi & Cohen, 2005
- Conditional Augmented Models
  - Layton, 2006 Thesis – Lattice C-Aug Chapter; Zhang, Ragni & Gales, 2010
- Segmental CRFs
  - Zweig & Nguyen, 2009

- These differ in
  - Where the Markov assumption is applied
  - What labels are available at training
    - Convexity of objective function
  - Definition of features

Approaches to Hidden Structure

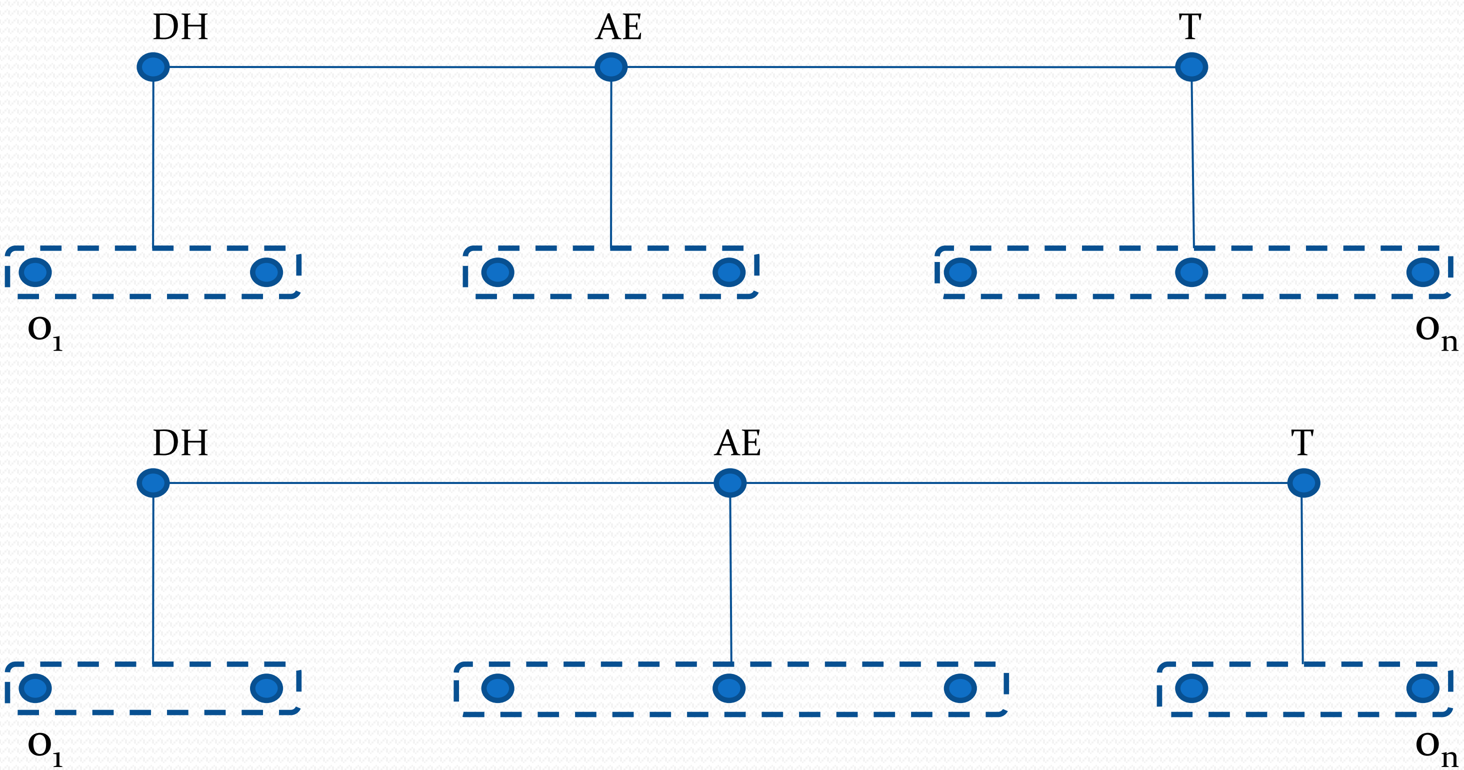
Method	Markov Assumption	Segmentation known in Training	Features Prescribed
HCRF	Frame level	No	No
Semi-Markov CRF	Segment	Yes	No
Conditional Augmented Models	Segment	No	Yes
Segmental CRF	Segment	No	No

One View of Structure



Consider all segmentations consistent with transcription / hypothesis
Apply Markov assumption at frame level to simplify recursions
Appropriate for frame level features

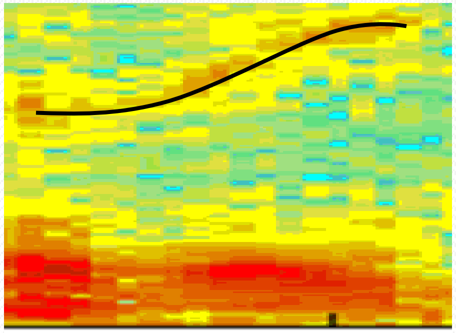
Another View of Structure



Consider all segmentations consistent with transcription / hypothesis
Apply Markov assumption at segment level only – "**Semi Markov**"
This means long-span segmental features can be used

Examples of Segment Level Features in ASR

- Formant trajectories
- Duration models
- Syllable / phoneme counts
- Min/max energy excursions
- Existence, expectation & levenshtein features described later



Examples of Segment Level Features in NLP

- Segment includes a name
- POS pattern within segment is DET ADJ N
- Number of capitalized words in segment
- Segment is labeled “Name” and has 2 words
- Segment is labeled “Name” and has 4 words
- Segment is labeled “Phone Number and has 7 words”
- Segment is labeled “Phone Number and has 8 words”

Is Segmental Analysis any Different?

- We are conditioning on all the observations
- Do we really need to hypothesize segment boundaries?
- YES – many features undefined otherwise:
  - Duration (of what?)
  - Syllable/phoneme count (count where?)
  - Difference in C_o between *start* and *end* of word

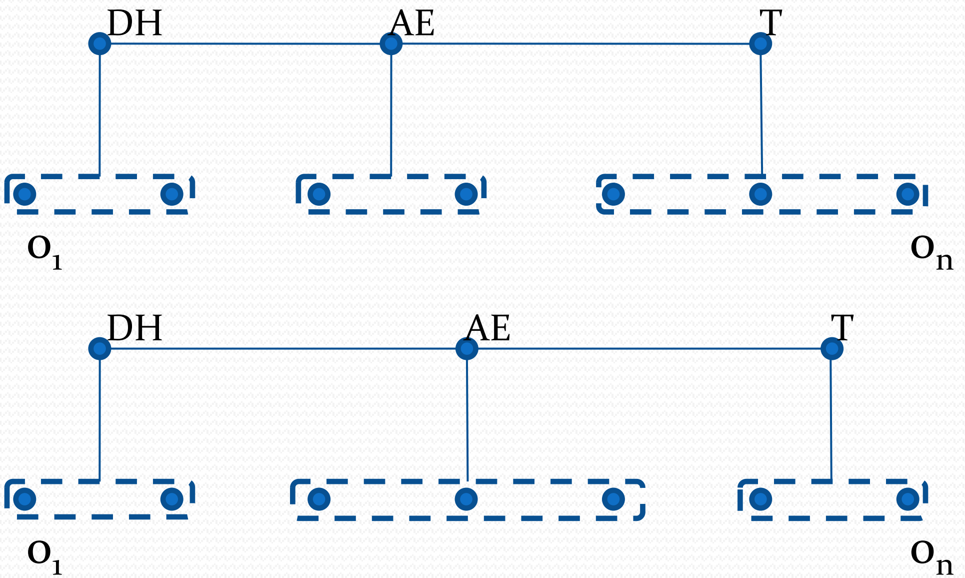
- Key Example: Conditional Augmented Statistical Models

Conditional Augmented Statistical Models

- Layton & Gales, "Augmented Statistical Models for Speech Recognition," ICASSP 2006
- As features use
  - Likelihood of *segment* wrt an HMM model
  - Derivative of likelihood wrt each HMM model parameter
- Frame-wise conditional independence assumptions of HMM are no longer present
- Defined *only* at *segment* level

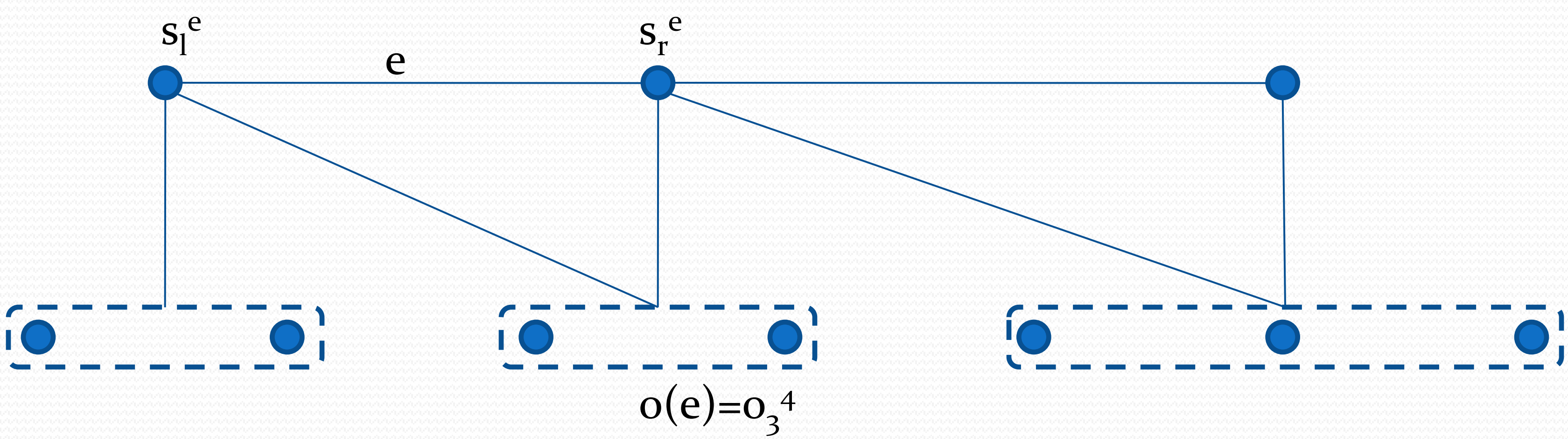
Now for Some Details

- Will examine general segmental case
- Then relate specific approaches

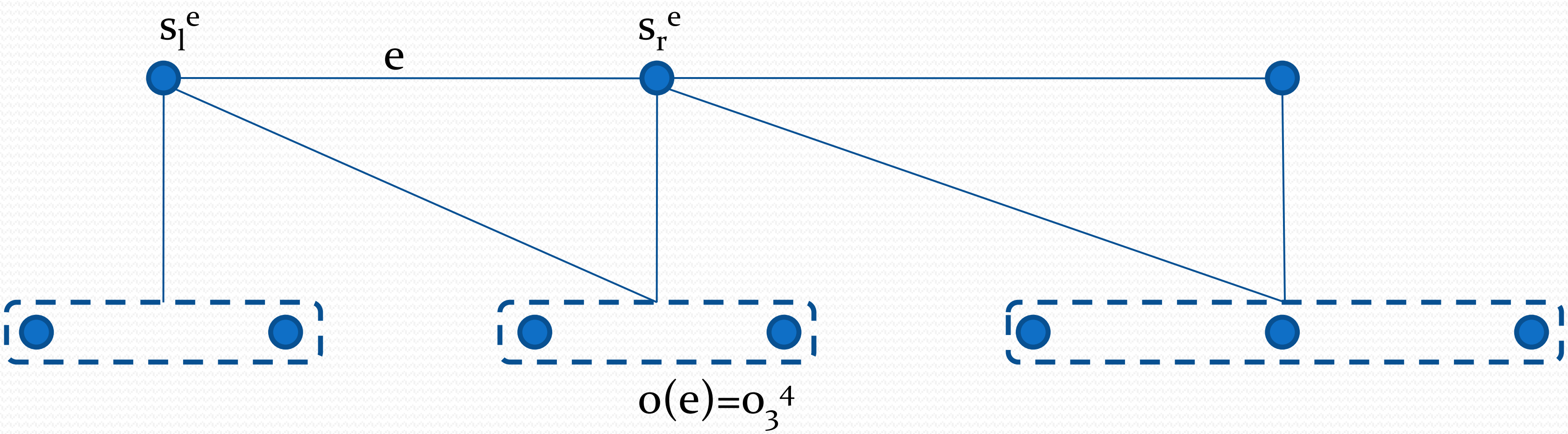


Segmental Notation & Fine Print

- We will consider feature functions that cover both transitions and observations
  - So a more accurate representation actually has diagonal edges
  - But we'll generally omit them for simpler pictures
- Look at a segmentation **q** in terms of its edges **e**
- s_l^e is the label associated with the left state on an edge
- s_r^e is the label associated with the right state on an edge
- O(e) is the span of observations associated with an edge



The Segmental Equations



$$P(\mathbf{s}\,|\,\mathbf{o}) = \frac{\sum_{\mathbf{q}\ st\ |\mathbf{q}|=|\mathbf{s}|} \exp\left(\sum_{e\in \mathbf{q},i} \lambda_i f_i(s_l^e, s_r^e, o(e))\right)}{\sum_{\mathbf{s}'} \sum_{\mathbf{q}\ st\ |\mathbf{q}|=|\mathbf{s}'|} \exp\left(\sum_{e\in \mathbf{q},i} \lambda_i f_i(s'^e_l, s'^e_r, o(e))\right)}$$

We must sum over all possible segmentations of the observations consistent with a hypothesized state sequence .

Conditional Augmented Model (Lattice version) in this View

$$P(\mathbf{s}\,|\,\mathbf{o}) = \frac{\sum_{\mathbf{q}\ st\ |\mathbf{q}|=|\mathbf{s}|} \exp\Big(\sum_{e\in\mathbf{q},i} \lambda_i f_i(s_l^e, s_r^e, o(e))\Big)}{\sum_{\mathbf{s}'} \sum_{\mathbf{q}\ st\ |\mathbf{q}|=|\mathbf{s}'|} \exp\Big(\sum_{e\in\mathbf{q},i} \lambda_i f_i(s'^e_l, s'^e_r, o(e))\Big)}$$

$$\exp\Big(\sum_{e\in\mathbf{q},i} \lambda_i f_i(s_l^e, s_r^e, o(e))\Big) \equiv \exp\Big(\sum_{e\in\mathbf{q}} \vec{\lambda}_{s_r^e} \big[L_{HMM(s_r^e)}(o(e))\ \ \nabla L_{HMM(s_r^e)}(o(e))\big]^T$$

Features precisely defined
HMM model likelihood
Derivatives of HMM model likelihood wrt HMM parameters

HCRF in this View

$$P(\mathbf{s}\,|\,\mathbf{o}) = \frac{\sum_{\mathbf{q}\ st\ |\mathbf{q}|=|\mathbf{s}|} \exp\left(\sum_{e\in\mathbf{q},i} \lambda_i f_i(s_l^e, s_r^e, o(e))\right)}{\sum_{\mathbf{s}'} \sum_{\mathbf{q}\ st\ |\mathbf{q}|=|\mathbf{s}'|} \exp\left(\sum_{e\in\mathbf{q},i} \lambda_i f_i(s'^e_l, s'^e_r, o(e))\right)}$$

$$\exp\left(\sum_{e\in\mathbf{q},i} \lambda_i f_i(s_l^e, s_r^e, o(e))\right) \equiv \exp\left(\sum_{k=1..N,\,i} \lambda_i f_i(s_{k-1}^e, s_k^e, o_k)\right)$$

Feature functions are decomposable at the *frame* level
Leads to simpler computations

Semi-Markov CRF in this View

$$P(\mathbf{s} \mid \mathbf{o}) = \frac{\sum_{\mathbf{q}\ st\ |\mathbf{q}|=|\mathbf{s}|} \exp\left(\sum_{e \in \mathbf{q}, i} \lambda_i f_i(s_l^e, s_r^e, o(e))\right)}{\sum_{\mathbf{s}'} \sum_{\mathbf{q}\ st\ |\mathbf{q}|=|\mathbf{s}'|} \exp\left(\sum_{e \in \mathbf{q}, i} \lambda_i f_i(s'^e_l, s'^e_r, o(e))\right)}$$

$$\sum_{\mathbf{q}\ st\ |\mathbf{q}|=|\mathbf{s}|} \exp\left(\sum_{e \in \mathbf{q}, i} \lambda_i f_i(s_l^e, s_r^e, o(e))\right) \equiv \exp\left(\sum_{e \in \mathbf{q}^*, i} \lambda_i f_i(s_l^e, s_r^e, o(e))\right)$$

A fixed segmentation is known at training
Optimization of parameters becomes convex

Structure Summary

- Sometimes only high-level information is available
  - E.g. the words someone said (training)
  - The words we think someone said (decoding)
- Then we must consider all the segmentations of the observations consistent with this
- HCRFs do this using a frame-level Markov assumption
- Semi-CRFs / Segmental CRFs do not assume independence between frames
  - Downside: computations more complex
  - Upside: can use segment level features
- Conditional Augmented Models prescribe a set of HMM based features

Break

Part 2: Algorithms

Key Tasks

- Compute optimal label sequence (decoding)

$$\arg\max_{s} P(s \mid o, \lambda)$$

- Compute likelihood of a label sequence

$$P(s \mid o, \lambda)$$

- Compute optimal parameters (training)

$$\arg\max_{\lambda} \prod_{d} P(s_d \mid o_d, \lambda)$$

Key Cases

Viterbi Assumption	Hidden Structure	Model
NA	NA	Log-linear classification
Frame-level	No	CRF
Frame-level	Yes	HCRF
Segment-level	Yes (decode only)	Semi-Markov CRF
Segment-level	Yes (train & decode)	C-Aug, Segmental CRF

Decoding

- The simplest of the algorithms
- Straightforward DP recursions

Viterbi Assumption	Hidden Structure	Model
NA	NA	Log-linear classification
Frame-level	No	CRF
Frame-level	Yes	HCRF
Segment-level	Yes (decode only)	Semi-Markov CRF
Segment-level	Yes (train & decode)	C-Aug, Segmental CRF

Cases we will go over

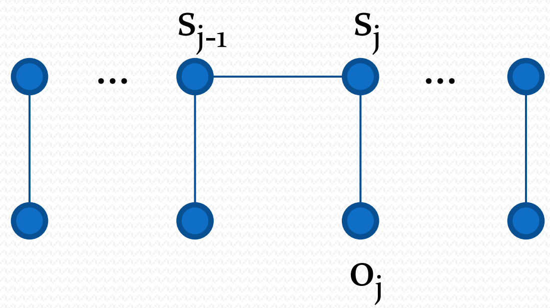
Flat log-linear Model

$$p(y \mid x) = \frac{\exp(\sum_i \lambda_i f_i(x, y))}{\sum_{y'} \exp(\sum_i \lambda_i f_i(x, y'))}$$

$$y^* = \arg\max_y \exp(\sum_i \lambda_i f_i(x, y))$$

Simply enumerate the possibilities and pick the best.

A Chain-Structured CRF



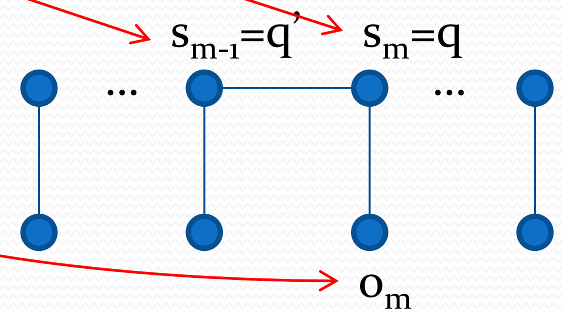
$$P(\mathbf{s}\,|\,\mathbf{o}) = \frac{\exp(\sum_j \sum_i \lambda_i f_i(s_{j-1}, s_j, o_j))}{\sum_{s'} \exp(\sum_j \sum_i \lambda_i f_i(s'_{j-1}, s'_j, o_j))}$$

$$\mathbf{s}^* = \arg\max_{\mathbf{s}} \; \exp(\sum_j \sum_i \lambda_i f_i(s_{j-1}, s_j, o_j))$$

Since **s** is a sequence there might be too many to enumerate.

Chain-Structured Recursions

The best way of getting here
is the best way of getting here
somehow and then making the
transition and accounting for
the observation



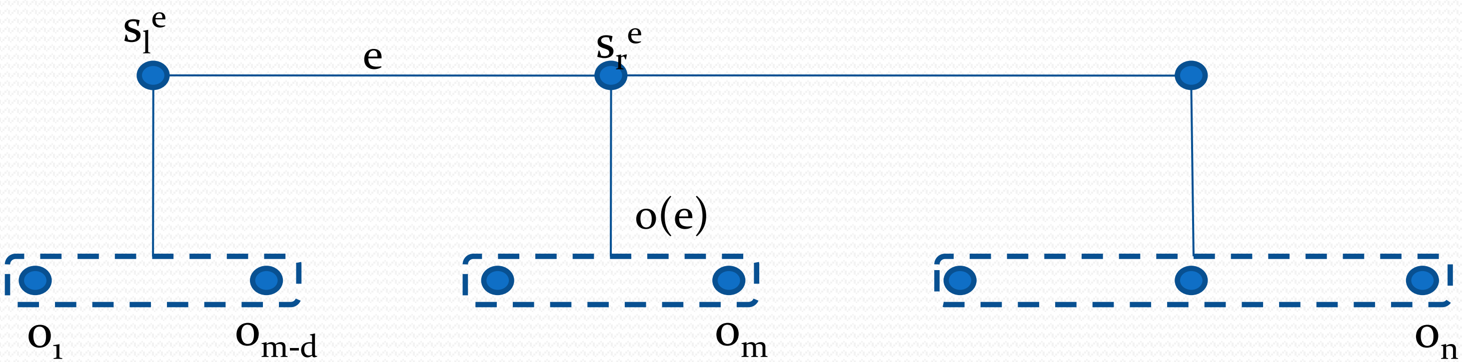
δ(m,q) is the best label sequence score that ends in position m with label q

$$\delta(m,q) = \arg\max_{q'} \delta(m-1, q')\exp(\sum_i \lambda_i f_i(q', q, o_m))$$

$$\delta(0,\bullet) = 1$$

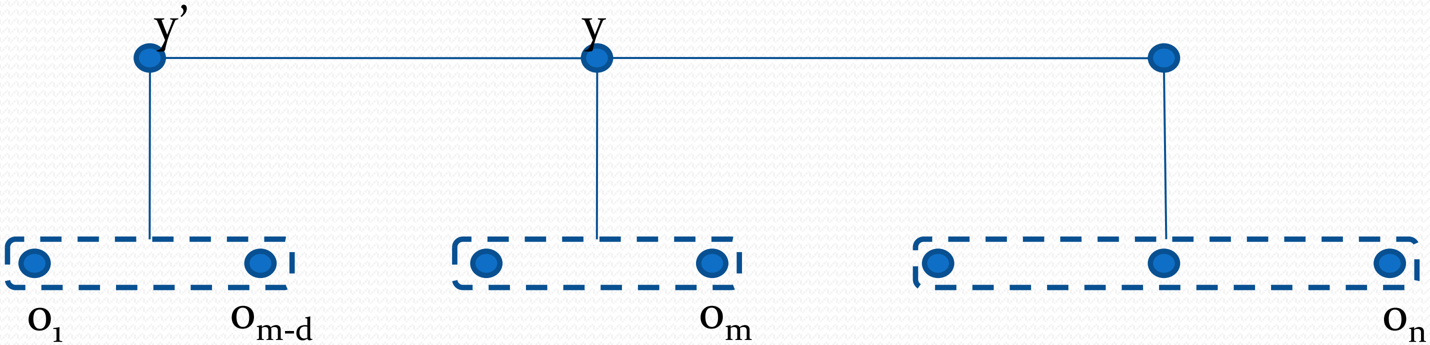
Recursively compute the δs
Keep track of the best q' decisions to recover the sequence

Segmental/Semi-Markov CRF



$$P(\mathbf{s}\,|\,\mathbf{o}) = \frac{\sum_{\mathbf{q}\ st\ |\mathbf{q}|=|\mathbf{s}|} \exp\left(\sum_{e\in\mathbf{q},i} \lambda_i f_i(s_l^e, s_r^e, o(e)\right)}{\sum_{\mathbf{s}'} \sum_{\mathbf{q}\ st\ |\mathbf{q}|=|\mathbf{s}'|} \exp\left(\sum_{e\in\mathbf{q},i} \lambda_i f_i(s'^e_l, s'^e_r, o(e)\right)}$$

Segmental/Semi-Markov Recursions



δ(*m*,*y*) is the best label sequence score that ends at observation *m* with state label *y*

$$\delta(m, y) = \arg\max_{y',\, d} \delta(m-d, y')\exp(\sum_i \lambda_i f_i(y', y, o_{m-d+1}^{m}))$$

$$\delta(0,\bullet) = 1$$

Recursively compute the δs
Keep track of the best q' *and* d decisions to recover the sequence

Computing Likelihood of a State Sequence

Viterbi Assumption	Hidden Structure	Model
NA	NA	Flat log-linear
Frame-level	No	CRF
Frame-level	Yes	HCRF
Segment-level	Yes (decode only)	Semi-Markov CRF
Segment-level	Yes (train & decode)	C-Aug, Segmental CRF

Cases we will go over

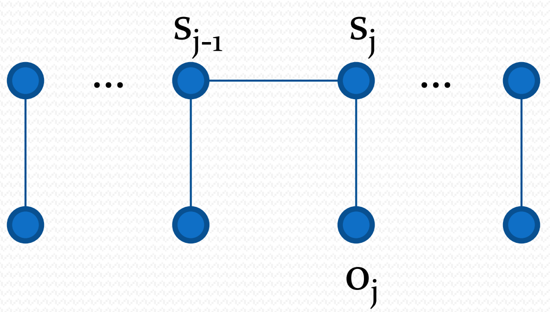
Flat log-linear Model

Plug in hypothesis

$$p(y \mid x) = \frac{\exp(\sum_i \lambda_i f_i(x, y))}{\sum_{y'} \exp(\sum_i \lambda_i f_i(x, y'))}$$

Enumerate the possibilities and sum.

A Chain-Structured CRF



Single hypothesis s
Plug in and compute

$$P(\mathbf{s}\,|\,\mathbf{o}) = \frac{\exp(\sum_j \sum_i \lambda_i f_i(s_{j-1}, s_j, o_j))}{\sum_{s'} \exp(\sum_j \sum_i \lambda_i f_i(s'_{j-1}, s'_j, o_j))}$$

Need a clever way of
summing over all hypotheses
To get normalizer Z

CRF Recursions

$$\alpha(m,q) = \sum_{\mathbf{s}_1^m \; st \; s_m = q} \exp \sum_{j=1..m} \sum_i \lambda_i f_i(s_{j-1}, s_j, o_j)$$

$\alpha(m,q)$ is the sum of the label sequence scores that end in position m with label q

$$\alpha(m,q) = \sum_{q'} \alpha(m-1,q') \exp(\sum_i \lambda_i f_i(q',q,o_m))$$

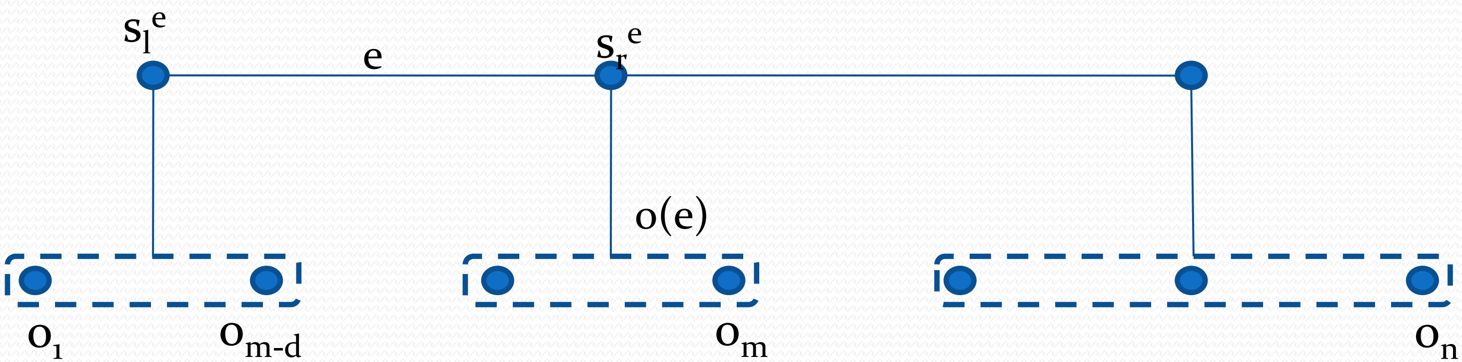
$$\alpha(0,\bullet) = 1$$

$$Z = \sum_{q'} \alpha(N,q')$$

Recursively compute the αs

Compute Z and plug in to find P(**s**|**o**)

Segmental/Semi-Markov CRF



$$P(\mathbf{s} \mid \mathbf{o}) = \frac{\displaystyle\sum_{\mathbf{q}\ st\ |\mathbf{q}|=|\mathbf{s}|} \exp\Big(\sum_{e \in \mathbf{q}, i} \lambda_i f_i(s_l^e, s_r^e, o(e))\Big)}{\displaystyle\sum_{\mathbf{s}'} \sum_{\mathbf{q}\ st\ |\mathbf{q}|=|\mathbf{s}'|} \exp\Big(\sum_{e \in \mathbf{q}, i} \lambda_i f_i(s'^{e}_l, s'^{e}_r, o(e))\Big)}$$

For segmental CRF numerator requires a summation too

Both Semi-CRF and segmental CRF require the same denominator sum

SCRF Recursions: Denominator

$$\alpha(m, y) = \sum_{\mathbf{s}\ st\ last(\mathbf{s})=y} \ \sum_{\mathbf{q}\ st\ |\mathbf{q}|=|\mathbf{s}|,\ last(\mathbf{q})=m} \exp \sum_{e \in \mathbf{q}, i} \lambda_i f_i(s_l^e, s_r^e, o(e))$$

a label a position

$\alpha(m,y)$ is the sum of the scores of all labelings and segmentations that end in position m with label y

$$\alpha(m, y) = \sum_{y'} \sum_{d} \alpha(m-d, y') \exp(\sum_i \lambda_i f_i(y', y, o_{m-d+1}^{m}))$$

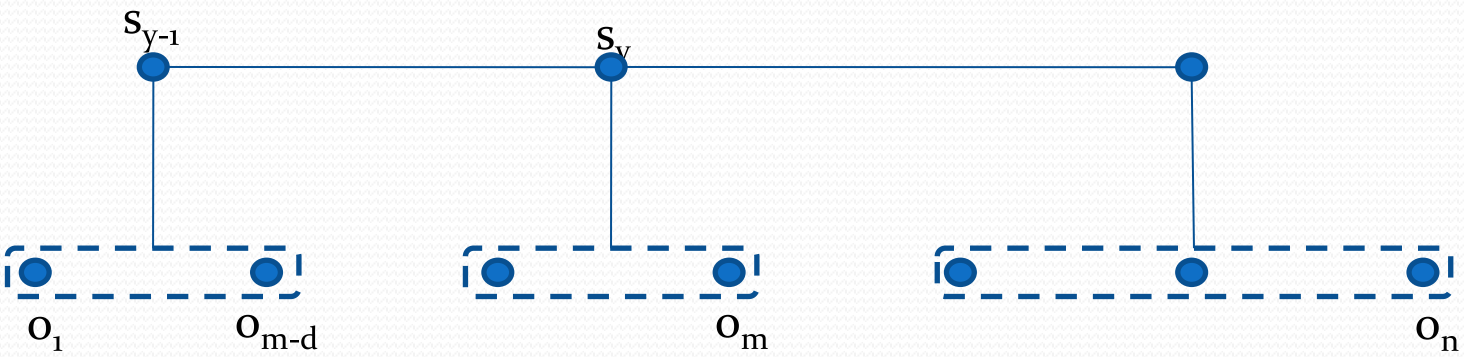
$$\alpha(0,\bullet) = 1$$

$$Z = \sum_{y'} \alpha(N, y')$$

Recursively compute the αs

Compute Z and plug in to find P(**s**|**o**)

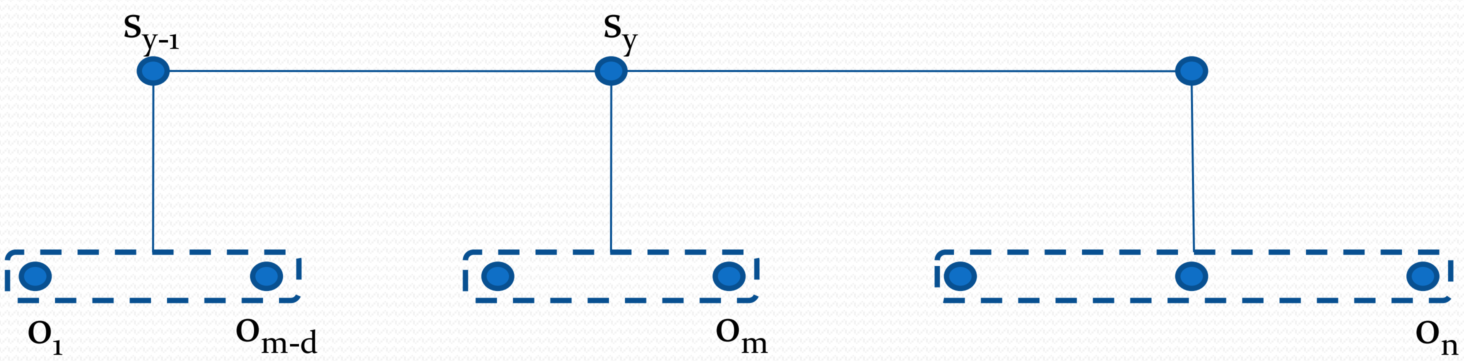
SCRF Recursions: Numerator



Recursion is similar with the state sequence fixed.
$\alpha^*(m,y)$ will now be the sum of the scores of all segmentations ending in an assignment of observation m to the y^{th} state.

Note the value of the y^{th} state is given!
y is now a positional index rather than state value.

Numerator (con't.)



$$\alpha^*(m,y) = \sum_{\mathbf{q}\ st\ |\mathbf{q}|=y} \exp \sum_{e\in\mathbf{q},i} \lambda_i f_i(s_l^e, s_r^e, o(e))$$

$$\alpha^*(m,y) = \sum_d \alpha^*(m-d, y-1)\exp(\sum_i \lambda_i f_i(s_{y-1}, s_y, o_{m-d+1}^m))$$

$$\alpha^*(0,\bullet) = 1$$

Note again that here y is the position into a given state sequence **s**

Summary: SCRF Probability

$$P(\mathbf{s}|\mathbf{o}) = \frac{\sum_{\mathbf{q}\ st\ |\mathbf{q}|=|\mathbf{s}|} \exp(\sum_{e\in\mathbf{q},i} \lambda_i f_i(s_l^e, s_r^e, o(e))}{\sum_{\mathbf{s}'} \sum_{\mathbf{q}\ st\ |\mathbf{q}|=|\mathbf{s}'|} \exp(\sum_{e\in\mathbf{q},i} \lambda_i f_i(s'^e_l, s'^e_r, o(e))}$$

$$= \frac{\alpha^*(N, |\mathbf{s}|)}{\sum_q \alpha(N, q)}$$

Compute alphas and numerator-constrained alphas with forward recursions
Do the division

Training

Viterbi Assumption	Hidden Structure	Model
NA	NA	Log-linear classification
Frame-level	No	CRF
Frame-level	Yes	HCRF
Segment-level	Yes (decode only)	Semi-Markov CRF
Segment-level	Yes (train & decode)	C-Aug, Segmental CRF

Will go over simplest cases. See also

- Gunawardana et al., Interspeech 2005 (HCRFs)
- Mahajan et al., ICASSP 2006 (HCRFs)
- Sarawagi & Cohen, NIPS 2005 (Semi-Markov)
- Zweig & Nguyen, ASRU 2009 (Segmental CRFs)

Training

- Specialized approaches
  - Exploit form of Max-Ent Model
    - Iterative Scaling (Darroch & Ratcliff, 1972)
      - $f_i(x,y) >= 0$ and $\Sigma_i f_i(x,y)=1$
    - Improved Iterative Scaling (Berger, Della Pietra & Della Pietra, 1996)
      - Only relies on non-negativity
- General approach: Gradient Descent
  - Write down the log-likelihood for one data sample
  - Differentiate it wrt the model parameters
  - Do your favorite form of gradient descent
    - Conjugate gradient
    - Newton method
    - R-Prop
  - Applicable regardless of convexity

Training with Multiple Examples

- When multiple examples are present, the contributions to the log-prob (and therefore gradient) are additive

$$L = \prod_j P(\mathbf{s}_j \mid \mathbf{o}_j)$$

$$\log L = \sum_j \log P(\mathbf{s}_j \mid \mathbf{o}_j)$$

  - To minimize notation, we omit the indexing and summation on data samples

Flat log-linear model

$$p(y \mid x) = \frac{\exp(\sum_i \lambda_i f_i(x,y))}{\sum_{y'} \exp(\sum_i \lambda_i f_i(x,y'))}$$

$$\log P(y \mid x) = \sum_i \lambda_i f_i(x,y) - \log \sum_{y'} \exp(\sum_i \lambda_i f_i(x,y'))$$

$$\frac{d}{d\lambda_k} \log P(y \mid x) = f_k(x,y) - \frac{\frac{d}{d\lambda_k} \sum_{y'} \exp(\sum_i \lambda_i f_i(x,y'))}{\sum_{y'} \exp(\sum_i \lambda_i f_i(x,y'))}$$

Flat log-linear Model Con't.

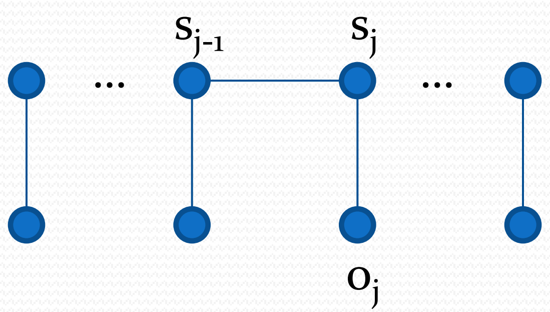
$$\frac{d}{d\lambda_k}\log P(y|x) = f_k(x,y) - \frac{\sum_{y'} \frac{d}{d\lambda_k} \exp(\sum_i \lambda_i f_i(x,y'))}{Z}$$

$$= f_k(x,y) - \frac{\sum_{y'} f_k(x,y')\exp(\sum_i \lambda_i f_i(x,y'))}{Z}$$

$$= f_k(x,y) - \sum_{y'} f_k(x,y')P(y'|x)$$

This can be computed by enumerating y'

A Chain-Structured CRF



$$P(\mathbf{s} \mid \mathbf{o}) = \frac{\exp\left(\sum_j \sum_i \lambda_i f_i(s_{j-1}, s_j, o_j)\right)}{\sum_{y'} \exp\left(\sum_j \sum_i \lambda_i f_i(s'_{j-1}, s'_j, o_j)\right)}$$

Chain-Structured CRF (con't.)

$$\log P(\mathbf{s}\,|\,\mathbf{o}) = \sum_j \sum_i \lambda_i f_i(s_{j-1}, s_j, o_j) - \log \sum_{s'} \exp(\sum_j \sum_i \lambda_i f_i(s'_{j-1}, s'_j, o_j))$$

$$\frac{d}{d\lambda_k} \log P(\mathbf{s}\,|\,\mathbf{o}) = \sum_j f_k(s_{j-1}, s_j, o_j)$$

$$- \frac{1}{Z} \sum_{s'} (\sum_j f_k(s'_{j-1}, s'_j, o_j)) \exp(\sum_j \sum_i \lambda_i f_i(s'_{j-1}, s'_j, o_j))$$

$$= \sum_j f_k(s_{j-1}, s_j, o_j) - \sum_{s'} \sum_j P(s'|o) f_k(s'_{j-1}, s'_j, o_j)$$

Easy to compute first term

Second is similar to the simple log-linear model, but:
* Cannot enumerate s' because it is now a sequence
* And must sum over positions j

Forward/Backward Recursions

$$\alpha(m,q) = \sum_{\mathbf{s}_1^m \, st \, s_m = q} \exp \sum_{j=1..m} \sum_i \lambda_i f_i(s_{j-1}, s_j, o_j)$$

$$= \sum_{q'} \alpha(m-1, q') \exp(\sum_i \lambda_i f_i(q', q, o_m))$$

$$\alpha(0,\bullet) = 1$$

$$Z = \sum_q \alpha(N, q)$$

$\alpha(m,q)$ is sum of partial path scores ending at position m, with label q (inclusive of observation m)

$$\beta(m,q) = \sum_{\mathbf{s}_m^N \, st \, s_m = q} \exp(\sum_{j=m+1..N} \sum_i \lambda_i f_i(s_{j-1,} s_j, o_j))$$

$$= \sum_{q'} \beta(m+1, q') \exp(\sum_i \lambda_i f_i(q, q', o_{m+1}))$$

$\beta(m,q)$ is sum of partial path scores starting at position m, with label q (exclusive of observation m)

Gradient Computation

$$\frac{d}{d\lambda_k}\log P(\mathbf{s}\,|\,\mathbf{o}) = \sum_j f_k(s_{j-1}, s_j, o_j)$$

$$-\frac{1}{Z}\sum_{s'}\Big(\sum_j f_k(s'_{j-1}, s'_j, o_j)\Big)\exp\Big(\sum_j\sum_i \lambda_i f_i(s'_{j-1}, s'_j, o_j)\Big)$$

$$= \sum_j f_k(s_{j-1}, s_j, o_j)$$

$$-\frac{1}{Z}\sum_j\sum_q\sum_{q'}\alpha(j,q)\beta(j+1,q')\exp\Big(\sum_i \lambda_i f_i(q,q',o_{j+1})\Big) f_k(q,q',o_{j+1})$$

1) Compute Alphas
2) Compute Betas
3) Compute gradient

Segmental Versions

- More complex; See
  - Sarawagi & Cohen, 2005
  - Zweig & Nguyen, 2009
- Same basic process holds
  - Compute alphas on forward recursion
  - Compute betas on backward recursion
  - Combine to compute gradient

Once We Have the Gradient

- Any gradient descent technique possible

1) Find a direction to move the parameters
   - Some combination of information from first and second derivative values
2) Decide how far to move in that direction
   - Fixed or adaptive step size
   - Line search
3) Update the parameter values and repeat

Conventional Wisdom

- Limited Memory BFGS often works well
  - Liu & Nocedal, Mathematical Programming (45) 1989
  - Sha & Pereira, HLT-NAACL 2003
  - Malouf, CoNLL 2002
- For HCRFs stochastic gradient descent and Rprop are as good or better
  - Gunawardana et al., Interspeech 2005
  - Mahajan, Gunawardana & Acero, ICASSP 2006
- Rprop is exceptionally simple

Rprop Algorithm

- Martin Riedmiller, “Rprop – Description and Implementation Details” Technical Report, January 1994, University of Karlsruhe.
- Basic idea:
  - Maintain a step size for each parameter
    - Identifies the “scale” of the parameter
  - See if the gradient says to increase or decrease the parameter
    - Forget about the exact value of the gradient
  - If you move in the same direction twice, take a bigger step!
  - If you flip-flop, take a smaller step!

Regularization

- In machine learning, often want to simplify models
  - Objective function can be changed to add a penalty term for complexity
    - Typically this is an L1 or L2 norm of the weight (lambda vector)
      - L1 leads to sparser models than L2
- For speech processing, some studies have found regularization
  - Necessary:
    L1-ACRFs by Hifny & Renals, Speech Communication 2009
  - Unnecessary if using weight averaging across time:
    Morris & Fosler-Lussier, ICASSP 2007

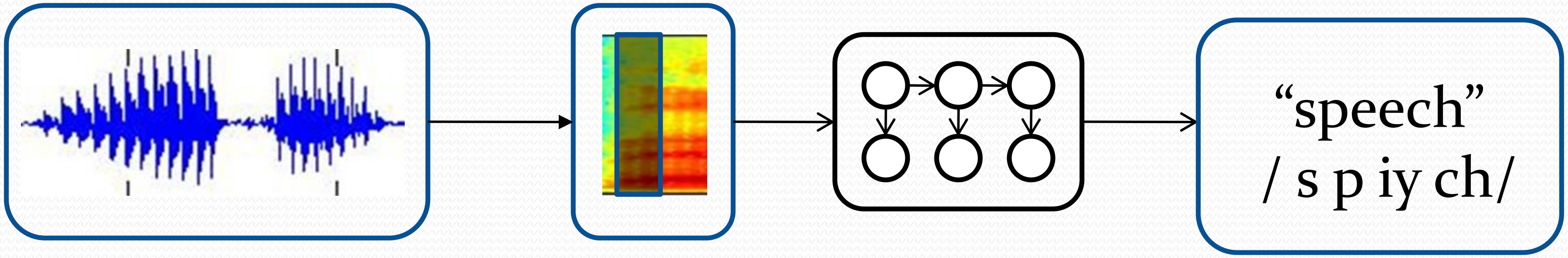
Case Studies (1)

CRF Speech Recognition with Phonetic Features

Acknowledgements to Jeremy Morris

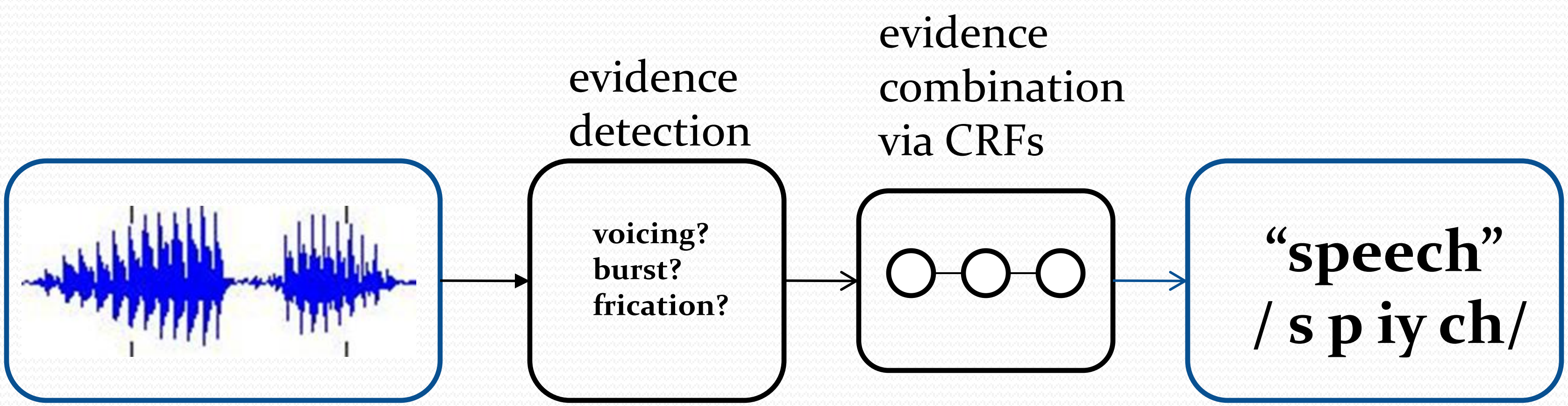
Top-down vs. bottom-up processing

- State-of-the-art ASR takes a top-down approach to this problem
  - Extract acoustic features from the signal
  - Model a process that generates these features
  - Use these models to find the word sequence that best fits the features



Bottom-up: detector combination

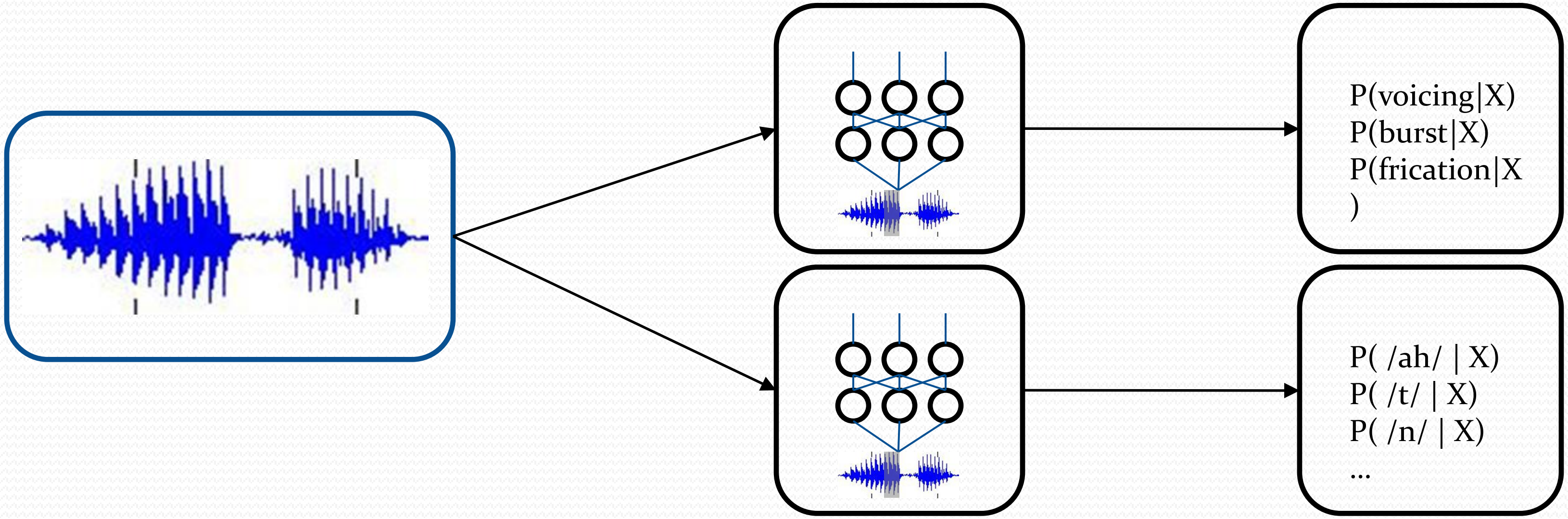
- A bottom-up approach using CRFs
  - Look for evidence of speech in the signal
    - Phones, phonological features
  - Combine this evidence together in log-linear model to find the most probable sequence of words in the signal



(Morris & Fosler-Lussier, 2006-2010)

Phone Recognition

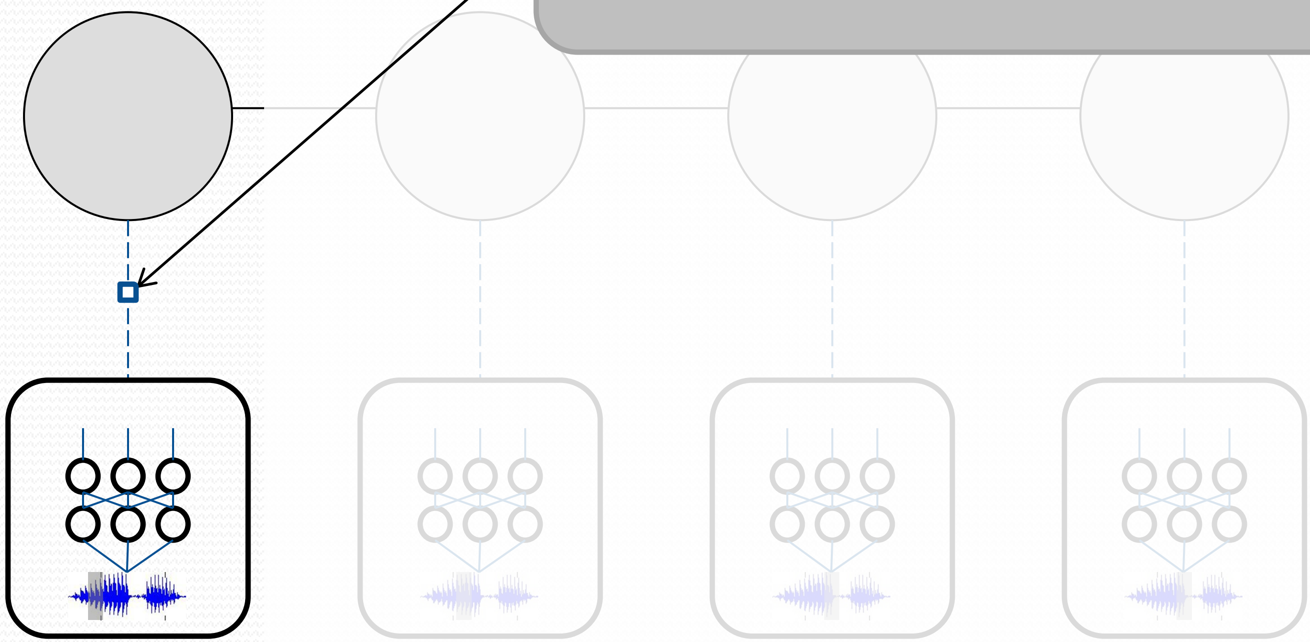
- What evidence do we have to combine?
  - MLP ANN trained to estimate frame-level posteriors for phonological features
  - MLP ANN trained to estimate frame-level posteriors for phone classes



Phone Recognition

- Use these MLP outputs to build *state feature functions*

$$s_{/t/,P(/t/|x)}(y,x) = \begin{cases} MLP_{P(/t/|x)}(x), & if \quad y = /t/ \\ 0, & otherwise \end{cases}$$

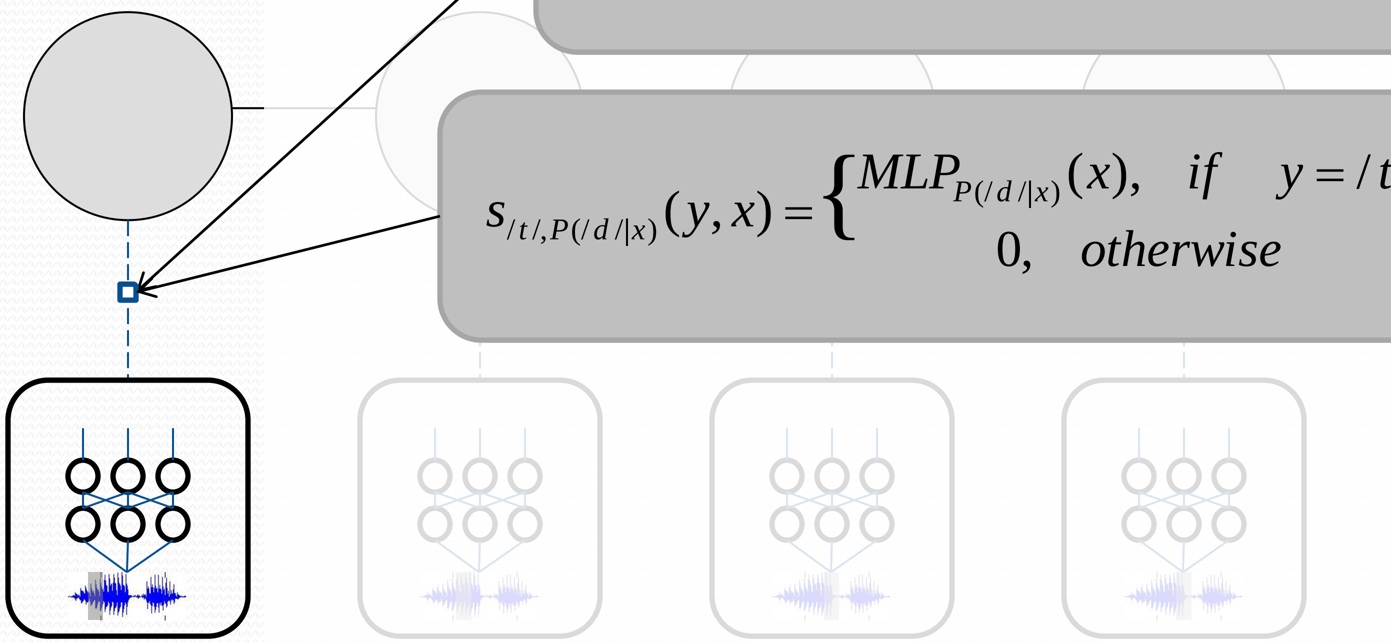


Phone Recognition

- Use these MLP outputs to build *state feature functions*

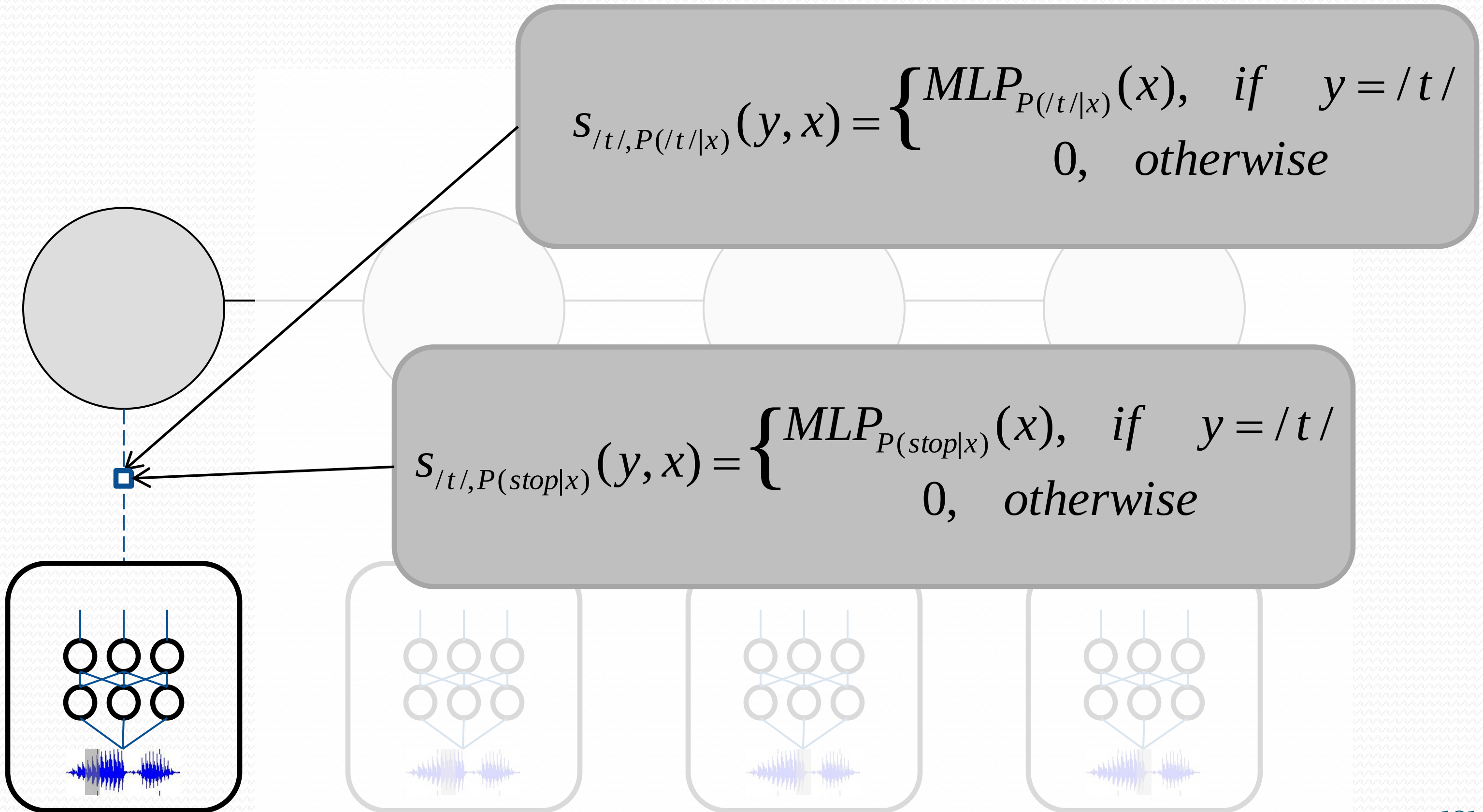
$$s_{/t/,P(/t/|x)}(y,x) = \begin{cases} MLP_{P(/t/|x)}(x), & if \quad y = /t/ \\ 0, & otherwise \end{cases}$$

$$s_{/t/,P(/d/|x)}(y,x) = \begin{cases} MLP_{P(/d/|x)}(x), & if \quad y = /t/ \\ 0, & otherwise \end{cases}$$



Phone Recognition

- Use these MLP outputs to build *state feature functions*



$$s_{/t/,P(/t/|x)}(y,x) = \begin{cases} MLP_{P(/t/|x)}(x), & if \quad y = /t/ \\ 0, & otherwise \end{cases}$$

$$s_{/t/,P(stop|x)}(y,x) = \begin{cases} MLP_{P(stop|x)}(x), & if \quad y = /t/ \\ 0, & otherwise \end{cases}$$

Phone Recognition

- Pilot task – phone recognition on TIMIT
  - ICSI Quicknet MLPs trained on TIMIT, used as inputs to the CRF models
  - Compared to Tandem and a standard PLP HMM baseline model
- Output of ICSI Quicknet MLPs as inputs
  - Phone class attributes (61 outputs)
  - Phonological features attributes (44 outputs)

Phone Recognition

Model	Accuracy
HMM (PLP inputs)	68.1%
CRF (phone classes)	70.2%
HMM Tandem16mix (phone classes)	70.4%
CRF (phone classes +phonological features)	**71.5%***
HMM Tandem16mix (phone classes+ phonological features)	70.2%

*Signficantly ($p<0.05$) better than comparable Tandem system (Morris & Fosler-Lussier 08)

What about word recognition?

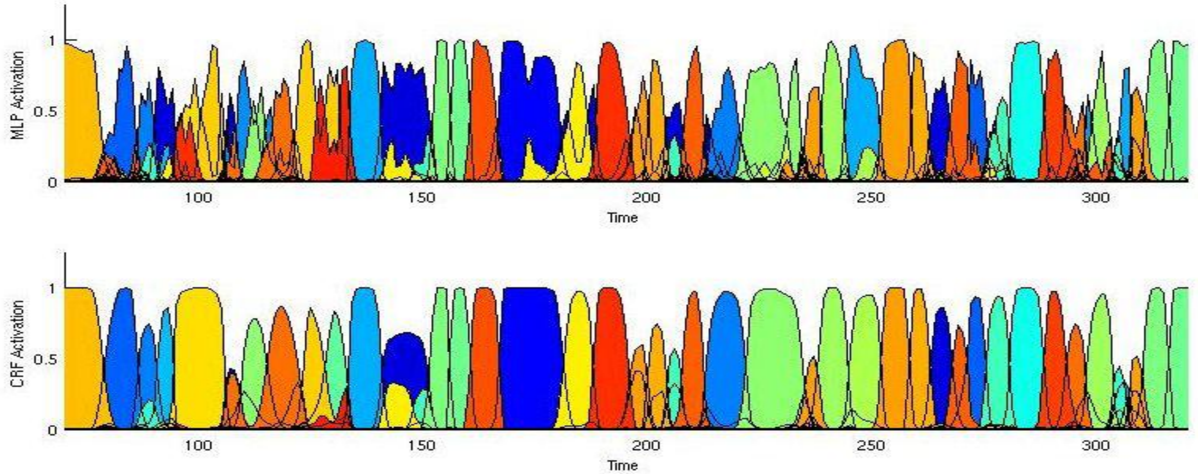
- CRF predicts phone labels for each frame
- Two methods for converting to word recognition:
  1. Use CRFs to generate local frame phone posteriors for use as features in an HMM (ala Tandem)
     - CRF + Tandem = CRANDEM
  2. Develop a new decoding mechanism for direct word decoding
     - More detail on this method

CRANDEM observations

- The Crandem approach worked well in phone recognition studies but did not immediately work as well as Tandem (MLP) for word recognition
  - Posteriors from CRF are smoother than MLP posteriors

MLP:

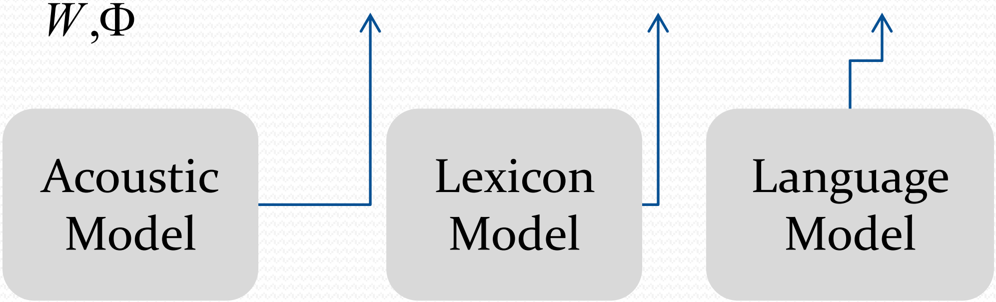
CRF:



  - Can improve Crandem performance by flattening the distribution

CRF Word Recognition

$$\arg\max_{W} P(W \mid O) \approx \arg\max_{W,\Phi} P(O \mid \Phi) P(\Phi \mid W) P(W)$$



- The standard model of ASR uses likelihood based acoustic models
- But CRFs provide a conditional acoustic model P(Φ|*O*)

CRF Word Recognition

$$\arg\max_{W} P(W \mid O) \approx \arg\max_{W,\Phi} \frac{P(\Phi \mid O)}{P(\Phi)} P(\Phi \mid W) P(W)$$

- CRF Acoustic Model
- Phone Penalty Model
- Lexicon Model
- Language Model

CRF Word Recognition

- Models implemented using OpenFST
  - Viterbi beam search to find best word sequence
- Word recognition on WSJ0
  - WSJ0 5K Word Recognition task
    - Same bigram language model used for all systems
  - Same MLPs used for CRF-HMM (Crandem) experiments
  - CRFs trained using 3-state phone model instead of 1-state model
  - Compare to original MFCC baseline (ML trained!)

CRF Word Recognition

Model	Dev WER	Eval WER
MFCC HMM reference	9.3%	8.7%
CRF (state only) – phone MLP input	11.3%	11.5%
CRF (state+trans) – phone MLP input	9.2%	8.6%
CRF (state+trans) – phone+phonological ftr MLPs input	8.3%	8.0%

NB: Eval improvement is not significant at $p<0.05$

- Transition features are important in CRF word decoding
- Combining features via CRF still improves decoding

Toolkit

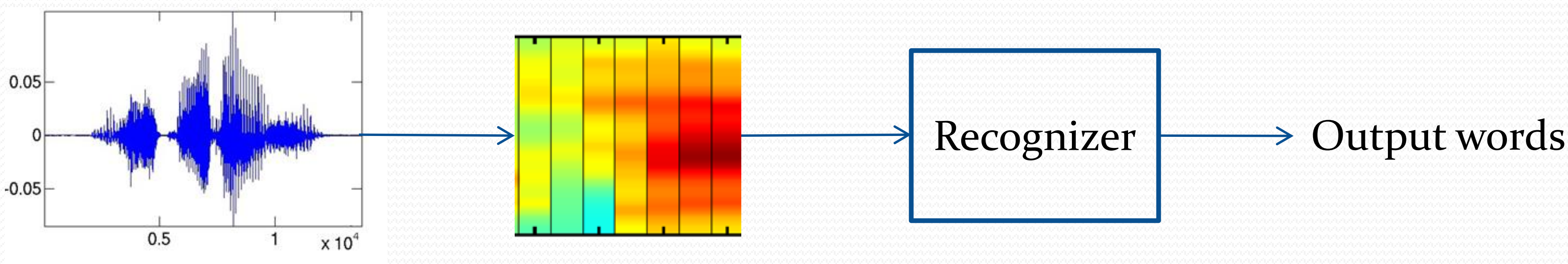
- The above experiments were done with the ASR-CRaFT toolkit, developed at OSU for the long sequences found in ASR
  - Primary author: Jeremy Morris
- Interoperable with the ICSI Quicknet MLP library
  - Uses same I/O routines
- Will be available from OSU Speech & Language Technology website
  - www.cse.ohio-state.edu/slate

Case Studies (2)

Speech Recognition with a Segmental CRF

The Problem

- State-of-the-art speech recognizers look at speech in just one way
  - Frame-by-frame
  - With one kind of feature



- And often the output is wrong

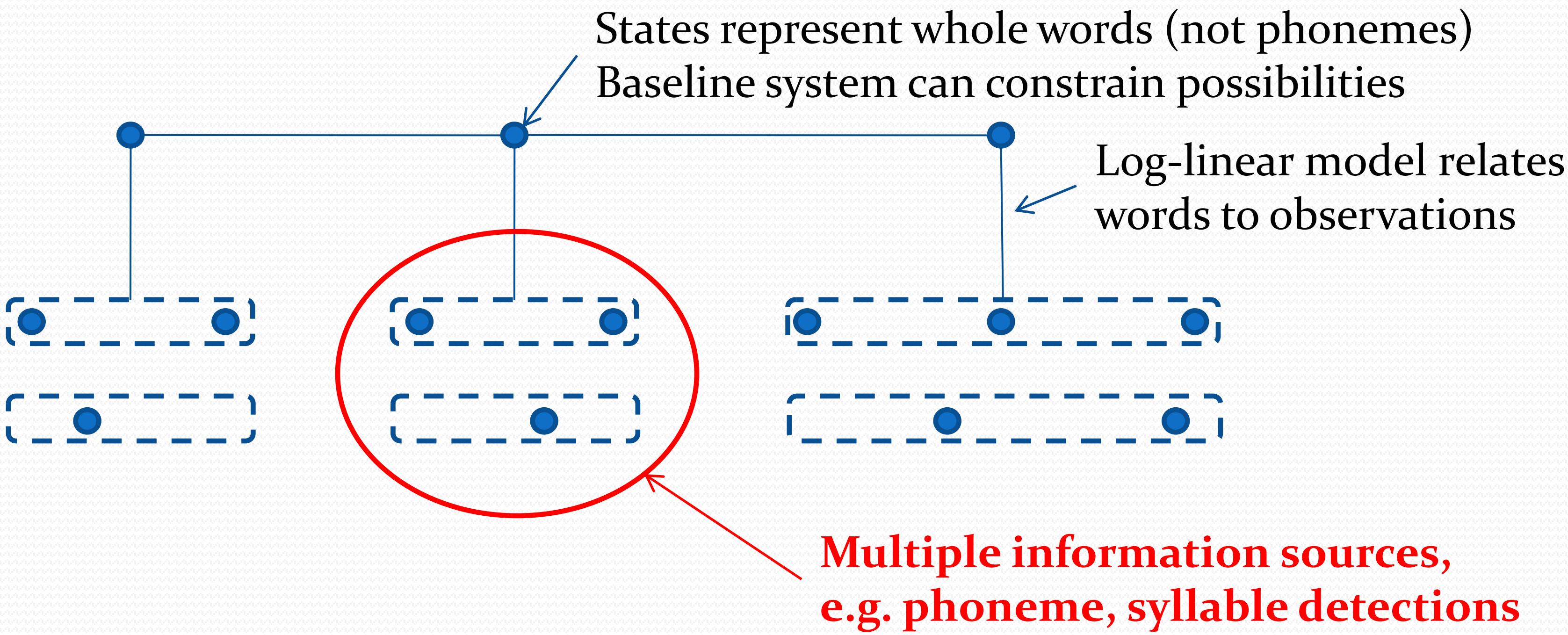
"Oh but he has a big challenge" ← **What we want (what was said)**

≠

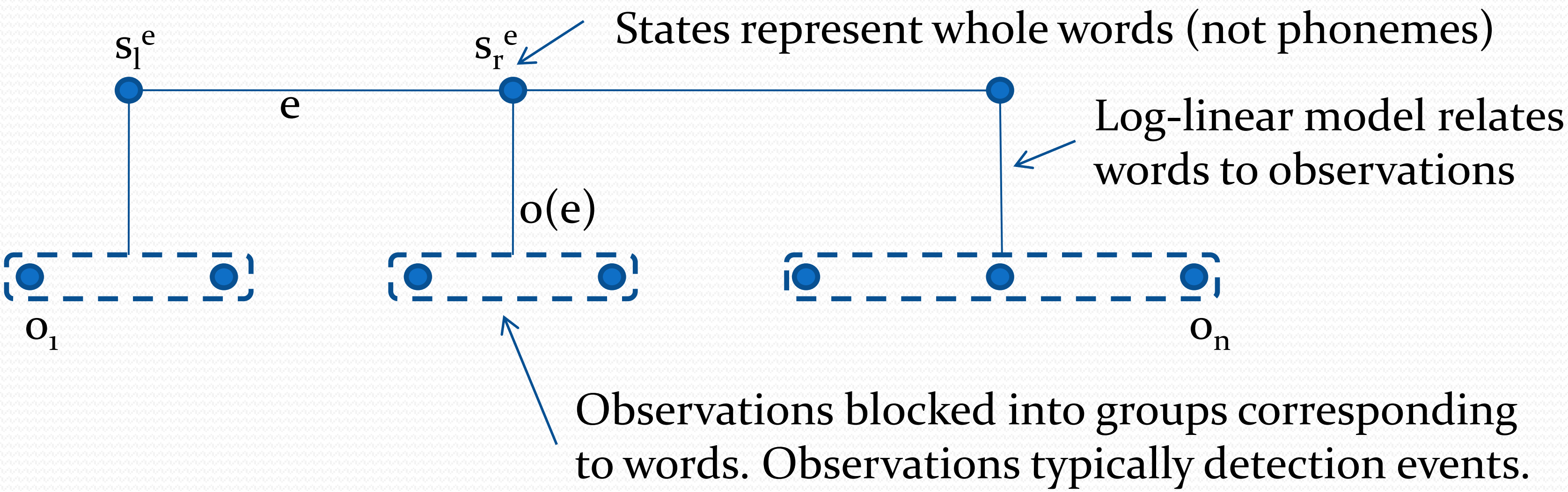
"ALREADY AS a big challenge" ← **What we get**

The Goal

- Look at speech in multiple ways
- Extract information from multiple sources
- Integrate them in a segmental, log-linear model



Model Structure

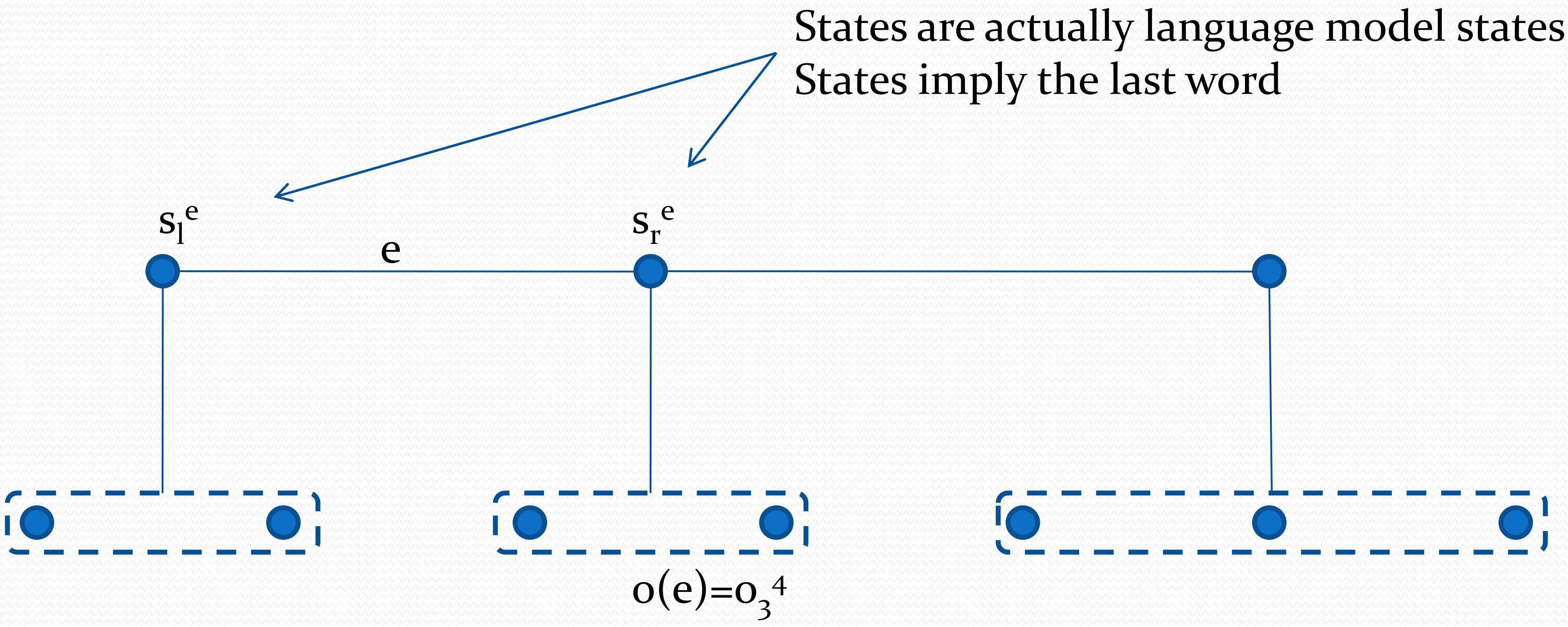


For a hypothesized word sequence s,
we must sum over all possible segmentations q of observations

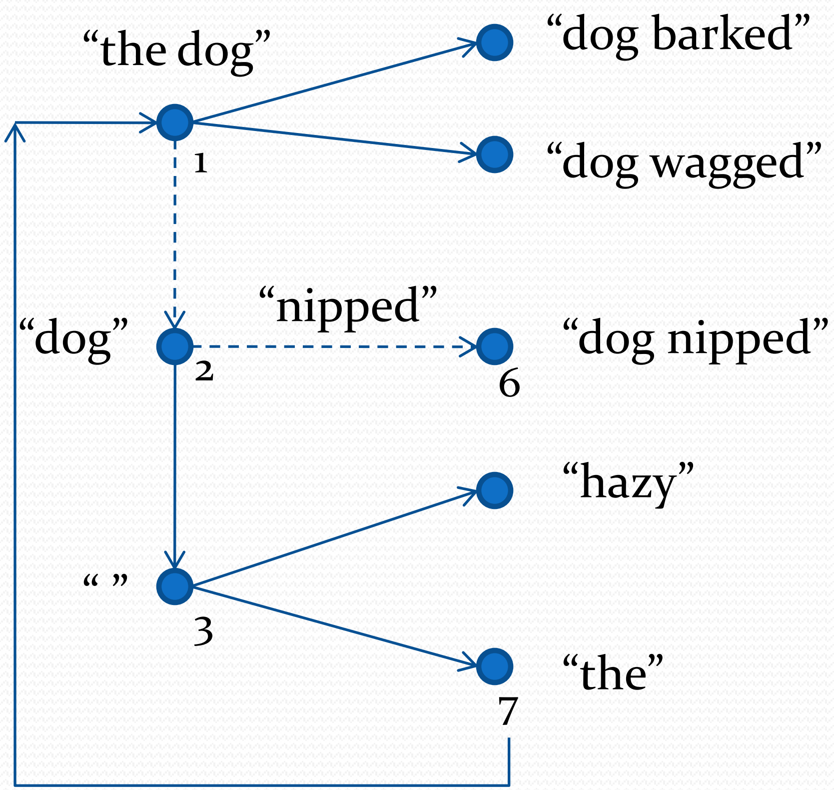
$$P(\mathbf{s}|\mathbf{o}) = \frac{\sum_{\mathbf{q}\ s.t.\ |\mathbf{q}|=|\mathbf{s}|} \exp(\sum_{e\in\mathbf{q},k} \lambda_k f_k(s_l^e, s_r^e, o(e)))}{\sum_{\mathbf{s}'} \sum_{\mathbf{q}\ s.t.\ |\mathbf{q}|=|\mathbf{s}'|} \exp(\sum_{e\in\mathbf{q},k} \lambda_k f_k(s_l'^e, s_r'^e, o(e)))}$$

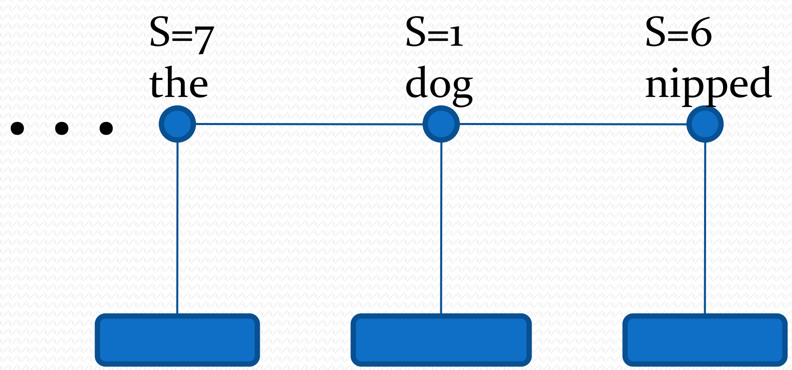
Training done to maximize product of label probabilities in the training data (CML).

The Meaning of States: ARPA LM



Embedding a Language Model





At minimum, we can use the state sequence to look up LM scores from the finite state graph. These can be features.

And we also know the actual arc sequence.

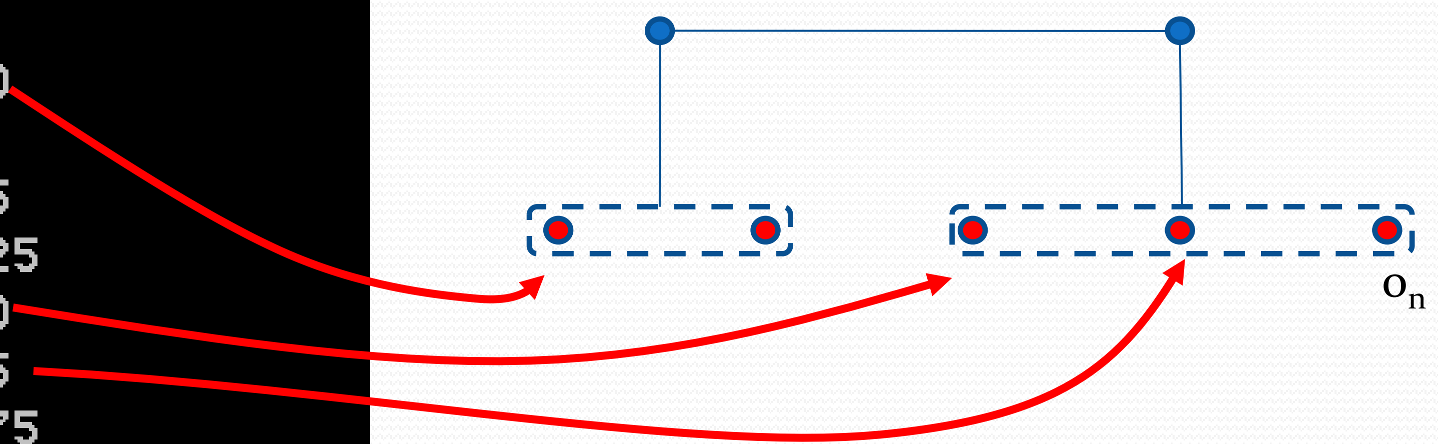
The SCARF Toolkit

- http://research.microsoft.com/en-us/projects/scarf/
- A toolkit which implements this model
- Talk on Thursday --
  - Zweig & Nguyen, "SCARF: A Segmental Conditional Random Field Toolkit for Speech Recognition" Interspeech 2010

Inputs (1)

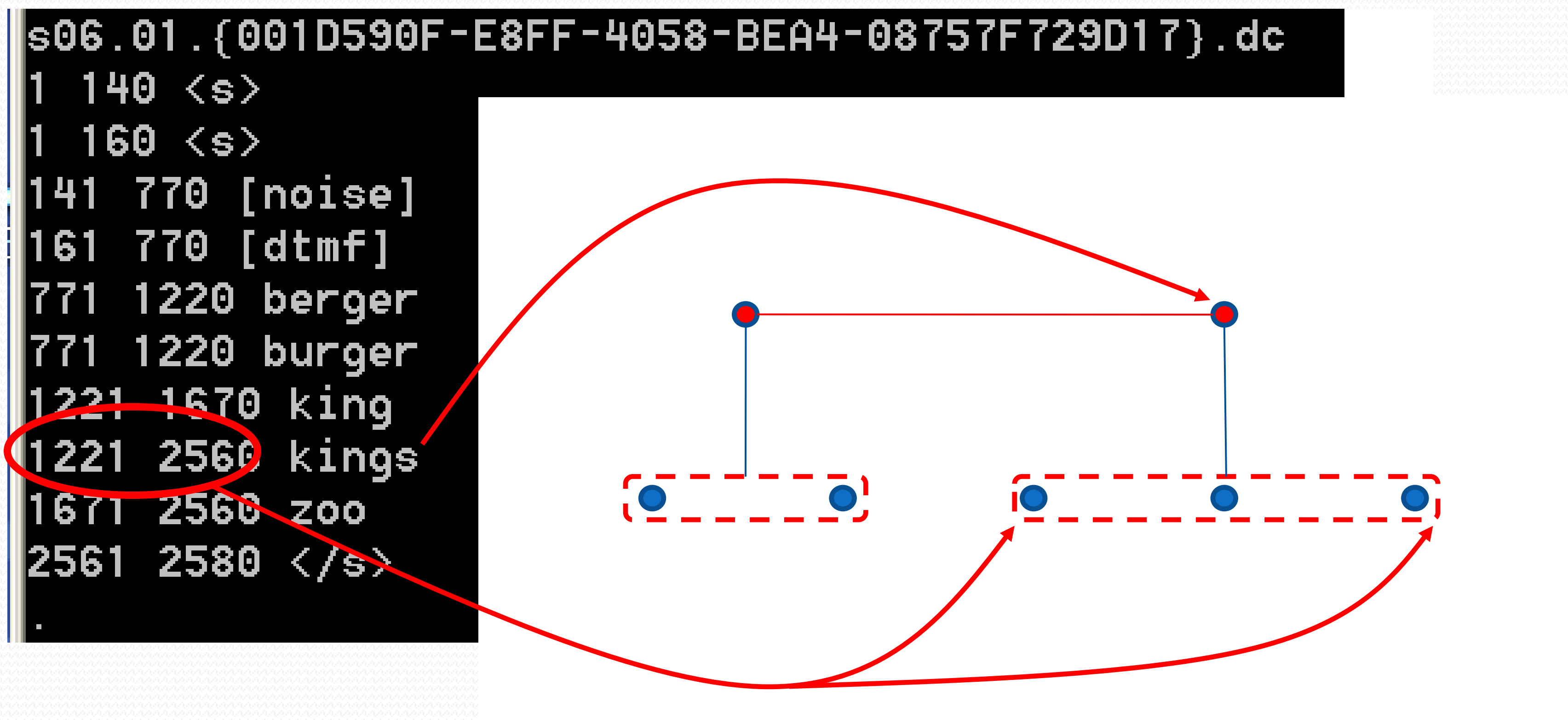
- Detector streams
  - (detection time) +
- Optional dictionaries
  - Specify the expected sequence of detections for a word

```
# phone stream
!sent_start 1
dtmf 460
b 790
er 880
r 980
g 1045
ax 1125
r 1210
z 1265
iy 1475
!sent_end 2580
```



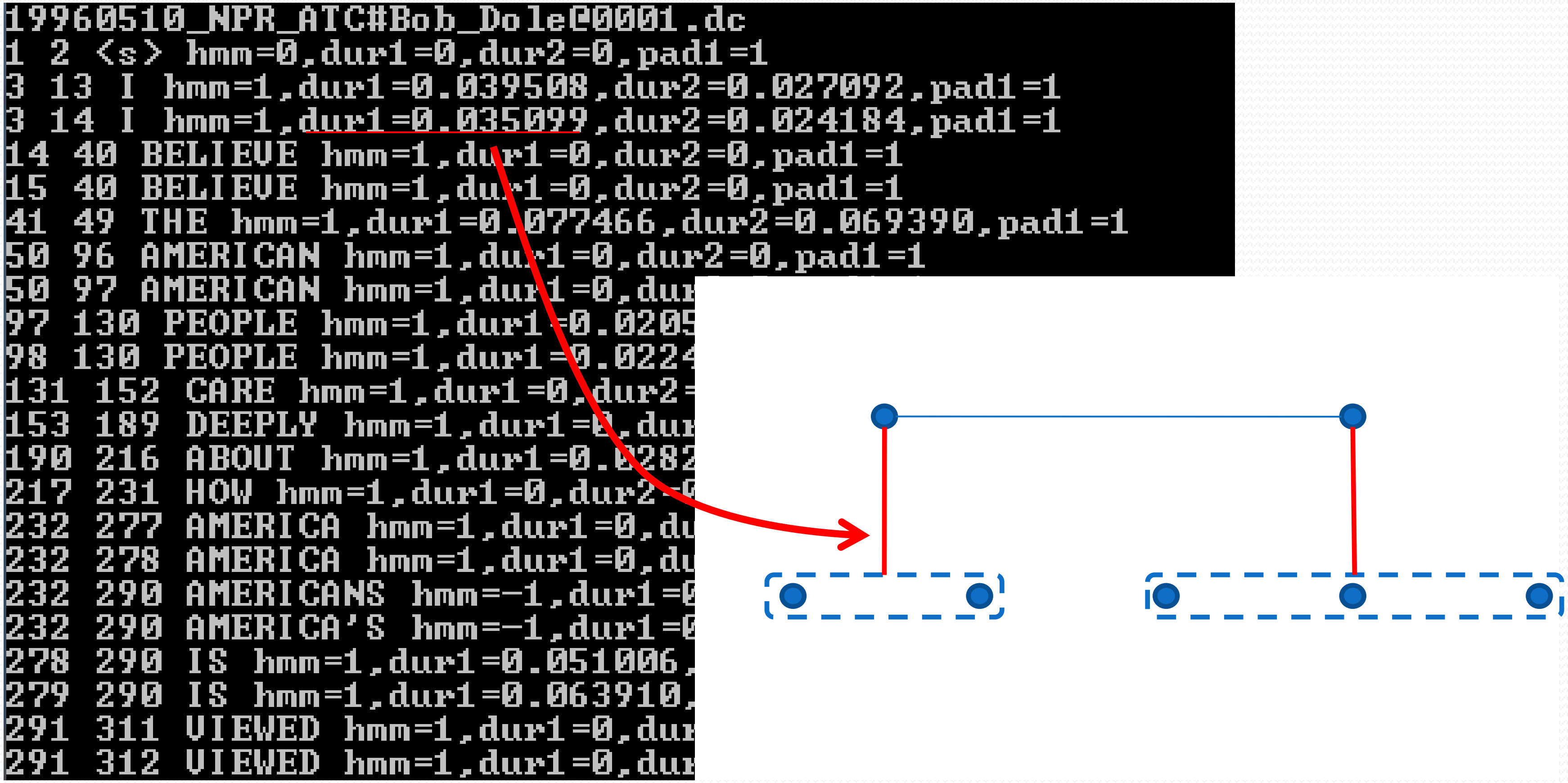
Inputs (2)

- Lattices to constrain search



Inputs (3)

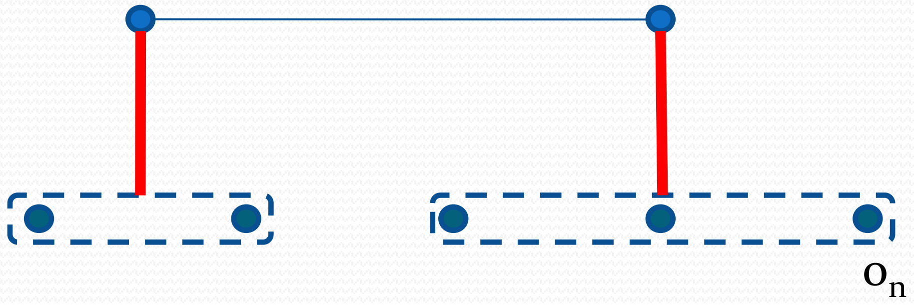
- User-defined features



```
19960510_NPR_ATC#Bob_Dole@0001.dc
1 2 <s> hmm=0,dur1=0,dur2=0,pad1=1
3 13 I hmm=1,dur1=0.039508,dur2=0.027092,pad1=1
3 14 I hmm=1,dur1=0.035099,dur2=0.024184,pad1=1
14 40 BELIEVE hmm=1,dur1=0,dur2=0,pad1=1
15 40 BELIEVE hmm=1,dur1=0,dur2=0,pad1=1
41 49 THE hmm=1,dur1=0.077466,dur2=0.069390,pad1=1
50 96 AMERICAN hmm=1,dur1=0,dur2=0,pad1=1
50 97 AMERICAN hmm=1,dur1=0,dur
97 130 PEOPLE hmm=1,dur1=0.0205
98 130 PEOPLE hmm=1,dur1=0.0224
131 152 CARE hmm=1,dur1=0,dur2=
153 189 DEEPLY hmm=1,dur1=0,dur
190 216 ABOUT hmm=1,dur1=0.0282
217 231 HOW hmm=1,dur1=0,dur2=0
232 277 AMERICA hmm=1,dur1=0,du
232 278 AMERICA hmm=1,dur1=0,du
232 290 AMERICANS hmm=-1,dur1=0
232 290 AMERICA'S hmm=-1,dur1=0
278 290 IS hmm=1,dur1=0.051006,
279 290 IS hmm=1,dur1=0.063910,
291 311 VIEWED hmm=1,dur1=0,dur
291 312 VIEWED hmm=1,dur1=0,dur
```

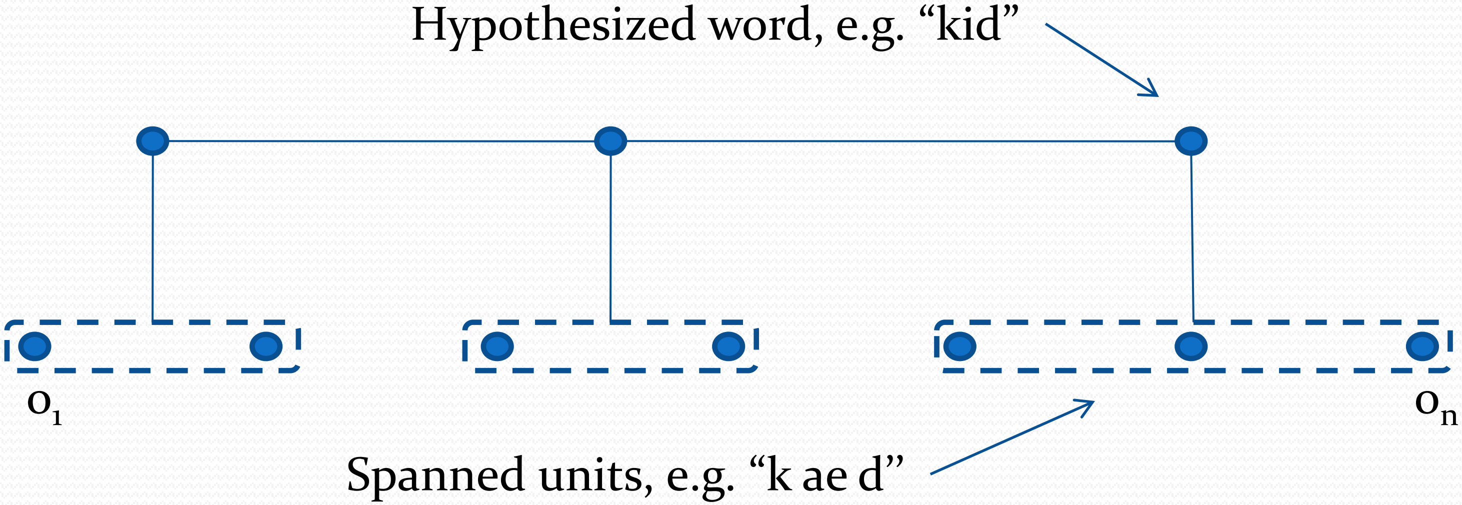
Detector-Based Features

- Array of features automatically constructed
- Measure forms of consistency between expected and observed detections
  - Differ in use of ordering information and generalization to unseen words
- Existence Features
- Expectation Features
- Levenshtein Features
- Baseline Feature



Existence Features

- Does unit X exist within the span of word Y?
- Created for all X,Y pairs in the dictionary and in the training data
- Can automatically be created for unit n-grams
- No generalization, but arbitrary detections OK



Expectation Features

- Use dictionary to get generalization ability across words!
- Correct Accept of u
  - Unit is in pronunciation of hypothesized word in dictionary, and it is detected in the span of the hypothesized word
  - ax k or d (dictionary pronunciation of accord)
  - ih k or (units seen in span)
- False Reject of u
  - Unit is in pronunciation of hypothesized word, but it is not in the span of the hypothesized word
  - ax k or d
  - ih k or
- False Accept of u
  - Unit is not in pronunciation of hypothesized word, and it is detected
  - ax k or d
  - ih k or
- Automatically created for unit n-grams

Levenshtein Features

- Match of u
- Substitution of u
- Insertion of u
- Deletion of u

Expected: ax k or d
Detected: ih k or *

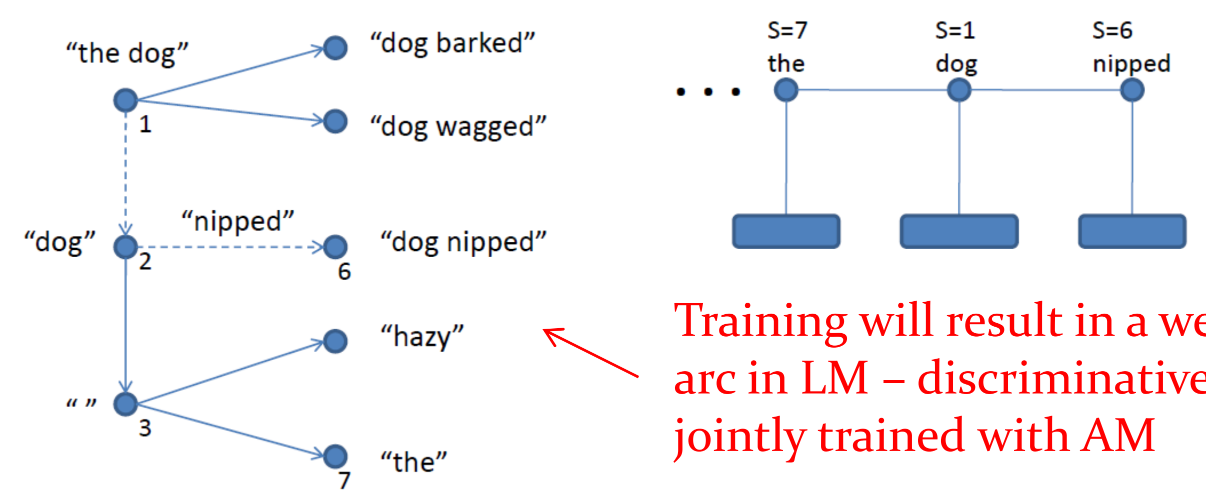
Sub-ax = 1
Match-k = 1
Match-or = 1
Del-d = 1

- Align the detector sequence in a hypothesized word's span with the dictionary sequence that's expected
- Count the number of each type of edits
- Operates only on the atomic units
- Generalization ability across words!

Language Model Features

- Basic LM:
  - Language model cost of transitioning between states.

- Discriminative LM training:
  - A binary feature for each arc in the language model
  - Indicates if the arc is traversed in transitioning between states

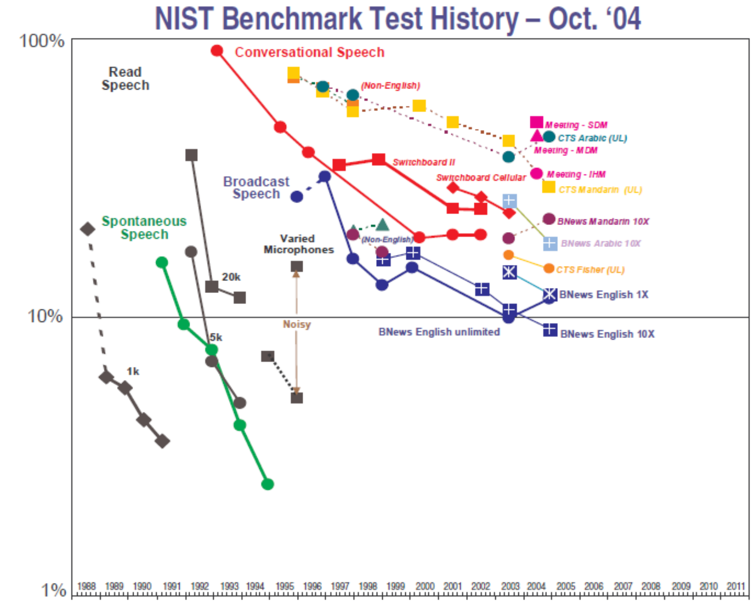


Training will result in a weight for each arc in LM – discriminatively trained, and jointly trained with AM

A Few Results from 2010 JHU Summer Workshop

Data Sets

- Wall Street Journal
  - Read newspaper articles
  - 81 hrs. training data
  - 20k open vocabulary test set
- Broadcast News
  - 430 hours training data
  - ~80k vocabulary
- World class baselines for both
  - 7.3% error rate WSJ (Leuven University)
  - 16.3% error rate BN (IBM Attila system)



Bottom Line Results

Wall Street Journal	WER	% Possible Gain
Baseline (SPRAAK / HMM)	7.3%	0%
+ SCARF, template features	**6.7**	**14**
(Lattice Oracle – best achievable)	2.9	100

Broadcast News	WER	% Possible Gain
Baseline (HMMw/ VTLN, HLDA, fMLLR, fMMI, mMMI, MLLR)	16.3%	0%
+ SCARF, word, phoneme detectors, scores	**15.0**	**25**
(Lattice Oracle – best achievable)	11.2	100

Case Studies (3)

A Sampling of NLP Applications

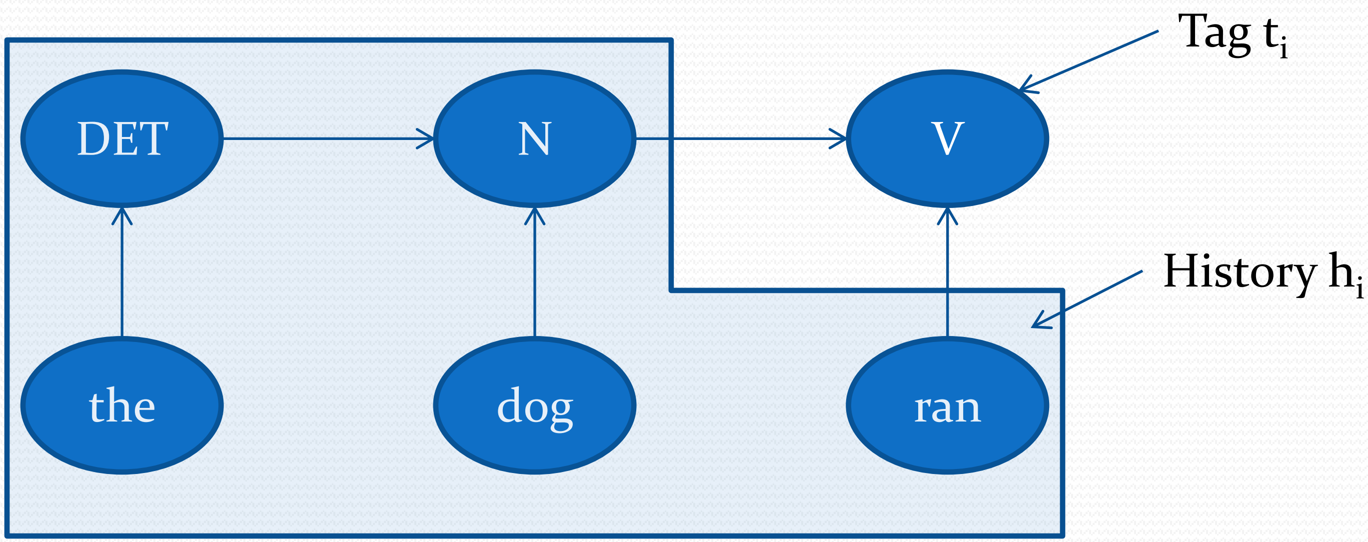
Intention of Case Studies

- Provide a sense of
  - Types of problems that have been tackled
  - Types of features that have been used
- Not any sort of extensive listing!
- Main point is ideas not experimental results (all good)

MEMM POS

- Reference: A. Ratnaparkhi, "A Maximum Entropy Model for Part-of-Speech Tagging," Proc. EMNLP, 1996
- Task: Part-of-Speech Tagging
- Model: Maximum Entropy Markov Model
  - Details Follow
- Features
  - Details Follow

MEMM POS Model



$$p(t_i \mid h_i) = \frac{\exp(\sum_j \lambda_j f_j(h_i, t_i))}{\sum_{t'} \exp(\sum_j \lambda_j f_j(h_i, t'))}$$

$$\mathbf{t}^* = \arg\max_{\mathbf{t}} \prod_i P(t_i \mid h_i)$$

Found via beam search

MEMM POS Features

Condition	Features	
w_i is not rare	$w_i = X$	& $t_i = T$
w_i is rare	X is prefix of w_i, $\|X\| \leq 4$	& $t_i = T$
	X is suffix of w_i, $\|X\| \leq 4$	& $t_i = T$
	w_i contains number	& $t_i = T$
	w_i contains uppercase character	& $t_i = T$
	w_i contains hyphen	& $t_i = T$
$\forall\ w_i$	$t_{i-1} = X$	& $t_i = T$
	$t_{i-2}t_{i-1} = XY$	& $t_i = T$
	$w_{i-1} = X$	& $t_i = T$
	$w_{i-2} = X$	& $t_i = T$
	$w_{i+1} = X$	& $t_i = T$
	$w_{i+2} = X$	& $t_i = T$

Voicemail Information Extraction

- Reference: Zweig, Huang & Padmanabhan, "Extracting Caller Information from Voicemail", Eurospeech 2001
- Task: Identify caller and phone number in voicemail
  - "Hi it's Peggy Cole Reed Balla's Secretary… reach me at x4567 Thanks"
- Model: MEMM
- Features:
  - Standard, plus class information:
    - Whether words belong to numbers
    - Whether a word is part of a stock phrase, e.g. "Talk to you later"

	Features		
$\forall w_i$	$w_i = X$	&	$t_i = T$
	$t_{i-1} = X$	&	$t_i = T$
	$t_{i-2}t_{i-1} = XY$	&	$t_i = T$
	$w_{i-2}w_{i-1} = XY$	&	$t_i = T$
	$w_{i-1}w_i = XY$	&	$t_i = T$
	$w_i w_{i+1} = XY$	&	$t_i = T$
	$w_{i+1}w_{i+2} = XY$	&	$t_i = T$

Shallow Parsing

- Reference: Sha & Pereira, “Shallow Parsing with Conditional Random Fields,” Proc. North American Chapter of ACL on HLT 2003
- Task: Identify noun phrases in text
  - **Rockwell** said **it** signed **a tentative agreement**.
  - Label each word as beginning a chunk (B), continuing a chunk (I), or external to a chunk (O)
- Model: CRF
- Features: Factored into transition and observation
  - See following overhead

Shallow Parsing Features

$q(y_{i-1}, y_i)$	$p(x, i)$
$y_i = y$ $y_i = y,\ y_{i-1} = y'$ $c(y_i) = c$	true
$y_i = y$ or $c(y_i) = c$	$w_i = w$ $w_{i-1} = w$ $w_{i+1} = w$ $w_{i-2} = w$ $w_{i+2} = w$ $w_{i-1} = w',\ w_i = w$ $w_{i+1} = w',\ w_i = w$ $t_i = t$ $t_{i-1} = t$ $t_{i+1} = t$ $t_{i-2} = t$ $t_{i+2} = t$ $t_{i-1} = t',\ t_i = t$ $t_{i-2} = t',\ t_{i-1} = t$ $t_i = t',\ t_{i+1} = t$ $t_{i+1} = t',\ t_{i+2} = t$ $t_{i-2} = t'',\ t_{i-1} = t',\ t_i = t$ $t_{i-1} = t'',\ t_i = t',\ t_{i+1} = t$ $t_i = t'',\ t_{i+1} = t',\ t_{i+2} = t$

$$f(y_{i-1}, y_i, \mathbf{x}, i) = p(\mathbf{x}, i)q(y_{i-1}, y_i)$$

Examples:

"The current label is 'OB' and the next word is "company".

"The current label is 'BI' and the POS of the current word is 'DET'"

Named Entity Recognition

- Reference: Sarawagi & Cohen, "Semi-Markov Conditional Random Fields for Information Extraction," NIPS 2005
- Task: NER
  - City/State from addresses
  - Company names and job titles from job postings
  - Person names from email messages
- Model: Semi-Markov CRF
- Features:
  - Word identity/position
  - Word capitalization
- Segmental Features:
  - Phrase presence
  - Capitalization *patterns* in segment
  - Combination non-segment features with segment initial/final indicator
  - Segment length

Whole Sentence Language Models

- Reference: Rosenfeld, Chen & Zhu, "Whole-Sentence Exponential Language Models: A Vehicle for Linguistic-Statistical Integration," Computer Speech & Language, 2001
- Task: Rescoring speech recognition nbest lists with a whole-sentence language model
- Model: Flat Maximum Entropy
- Features:
  - Word ngrams
  - Class ngrams
  - Leave-one-out ngrams (skip ngrams)
  - Presence of constituent sequences in a parse

Conclusions

Tutorial summary

- Provided an overview of direct models for classification and sequence recognition
  - MaxEnt, MEMM, (H)CRF, Segmental CRFs
  - Training & recognition algorithms
  - Case studies in speech & NLP
- Fertile area for future research
  - Methods are flexible enough to incorporate different representation strategies
  - Toolkits are available to start working with ASR or NLP problems

Future Research Directions

- Feature design for ASR – have only scratched the surface of different acoustic representations
- Feature induction – MLPs induce features using hidden nodes, can look at backprop methods for direct models
  - Multilayer CRFs (Prabhavalkar & Fosler-Lussier 2010)
  - Deep Belief Networks (Hinton, Osindero & Teh 2006)
- Algorithmic design
  - Exploration of Segmentation algorithms for CRFs
- Performance Guarantees

References

Berger, Della Pietra & Della Pietra, A maximum entropy approach to natural language processing," Computational Linguistics, 1996.

Darroch & Ratcliff, "Generalized iterative scaling for log-linear models," The Annals of Mathematical Statistics, 1972.

Gunawardana, Mahajan, Acero & Platt, "Hidden Conditional Random Fields for Phone Classification," Interspeech 2005

Hammersley & Clifford, "Markov fields on finite graphs and lattices," unpublished manuscript, 1971.

Heigold et al., "A Flat Direct Model for Speech Recognition," ICASSP 2009.

Hifny & Renals, "Speech recognition using augmented conditional random fields," IEEE Trans. Acoustics, Speech, and Language Processing, 2009.

Hinton, Osindero & Teh, "A fast learning algorithm for deep belief nets. Neural Computation," Neural Computation, 2006.

Kuo & Gao, "Maximum Entropy Direct Models for Speech Recognition," IEEE Trans. Speech and Audio Processing, 2006

Layton, "Augmented statistical models for classifying sequence data," Ph.D. Thesis, Cambridge U., 2006.

Layton & Gales, "Augmented Statistical Models for Speech Recognition," ICASSP 2006

Lafferty, McCallum & Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," Proc ICML 2001

Liu & Nocedal, "On the limited memory BFGS method for large scale optimization," Mathematical programming, 1989.

Malouf, "Markov models for language-independent named entity recognition," CoNLL 2002.

Morris, "Conditional Random Fields for Automatic Speech Recognition," Ph.D. Thesis, The Ohio State University, 2010.

Morris & Fosler-Lussier, "CRANDEM: Conditional Random Fields for Word Recognition," Interspeech 2009.

Morris & Fosler-Lussier, "Conditional Random Fields for Integrating Local Discriminative Classifiers," IEEE Trans. Acoustics, Speech, and Language Processing, 2010.

Mahajan, Gunawardana & Acero, "Training Algorithms for Hidden Conditional Random Fields," ICASSP 2006

McCallum, Freitag & Pereira, "Maximum Entropy Markov Models for Information Extraction and Segmentation," Proc. ICML 2000.

Prabhavalkar & Fosler-Lussier, "Backpropagation Training for Multilayer Conditional Random Field Based Phone Recognition," ICASSP 2010.

Ratnaparkhi, "A Maximum Entropy Model for Part-of-Speech Tagging," Proc. EMNLP, 1996

Riedmiller, “Rprop – Description and Implementation Details” Technical Report, University of Karlsruhe, 1994.

Rosenfeld, Chen & Zhu, “Whole-Sentence Exponential Language Models: A Vehicle for Linguistic-Statistical Integration,” Computer Speech & Language, 2001

Sarawagi & Cohen, “Semi-Markov Conditional Random Fields for Information Extraction,” NIPS 2005

Sha & Pereira, “Shallow Parsing with Conditional Random Fields,” Proc. NAACL-HLT 2003

Zhang, Ragni & Gales, “Structured Log Linear Models for Noise Robust Speech Recognition,” http://svr-www.eng.cam.ac.uk/~mjfg/zhang10.pdf 2010

Zweig, Huang & Padmanabhan, “Extracting Caller Information from Voicemail”, Eurospeech 2001

Zweig & Nguyen, “A Segmental CRF Approach to Large Vocabulary Continuous Speech Recognition,” ASRU 2009

Zweig & Nguyen, “SCARF: A Segmental Conditional Random Field Toolkit for Speech Recognition,” Proc. Interspeech 2010

Other good overviews of CRFs:

Sutton & McCallum, “An Introduction to Conditional Random Fields for Relational Learning” In Getoor & Taskar, editors. *Introduction to Statistical Relational Learning*. 2007.

Wallach, "Conditional Random Fields: An Introduction." Technical Report MS-CIS-04-21. Department of Computer and Information Science, University of Pennsylvania, 2004.