

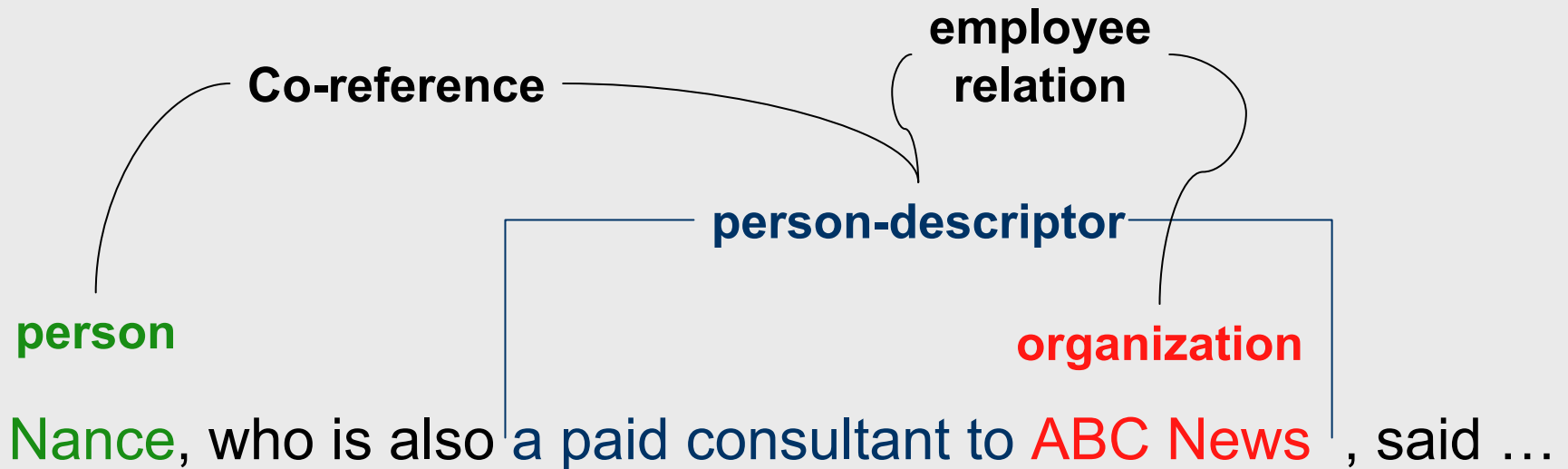
Undirected Graphical Models for Sequence Analysis

Fernando Pereira

University of Pennsylvania

Joint work with John Lafferty,
Andrew McCallum, and Fei Sha

Motivation: Information Extraction



- Information extraction as labeling:
 - States represent (parts of) entities
 - Relations
 - Cascaded labelers
 - Structured labels

Additional Motivation

- Tagging
- Shallow parsing
- Biological sequence analysis: combine
 - Local statistics: “content sensors”
 - “Signaling”: translation initiation, splice sites

Global Labeling

- Train to minimize labeling loss

$$\hat{\Lambda} = \operatorname{argmin}_{\Lambda} \sum_i \operatorname{Loss}(\mathbf{o}_i, \mathbf{s}_i \mid \Lambda)$$

- Computing the best labeling:

$$\operatorname{argmin}_{\mathbf{s}} \operatorname{Loss}(\mathbf{o}, \mathbf{s} \mid \hat{\Lambda})$$

- Efficient minimization requires:

- A common currency for local labeling decisions
- Efficient algorithm to combine the decisions

Local Labeling

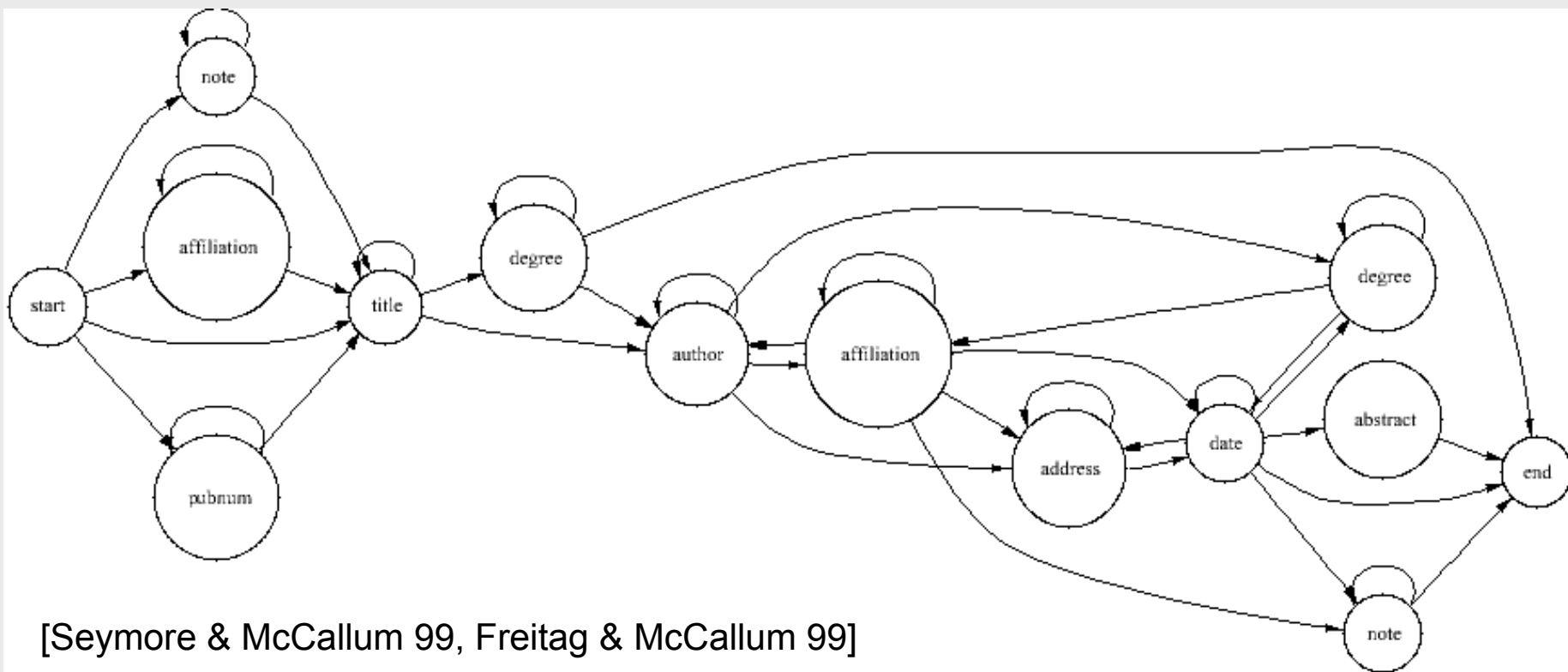
- Train to minimize the per-decision loss in context

$$\hat{\Lambda} = \operatorname{argmin}_{\Lambda} \sum_i \sum_{0 \leq j < |\mathbf{o}_i|} \operatorname{loss}(s_{i,j} \mid \mathbf{o}_i, \mathbf{s}_i^{(j)}; \Lambda)$$

- Apply by guessing context and finding each lowest-loss label:

$$\operatorname{argmin}_{s_j} \operatorname{loss}(s_j \mid \mathbf{o}, \hat{\mathbf{s}}^{(j)}; \hat{\Lambda})$$

Information Extraction with HMMs



- Observations: words
- States correspond to fields to extract

Problems with HMMs

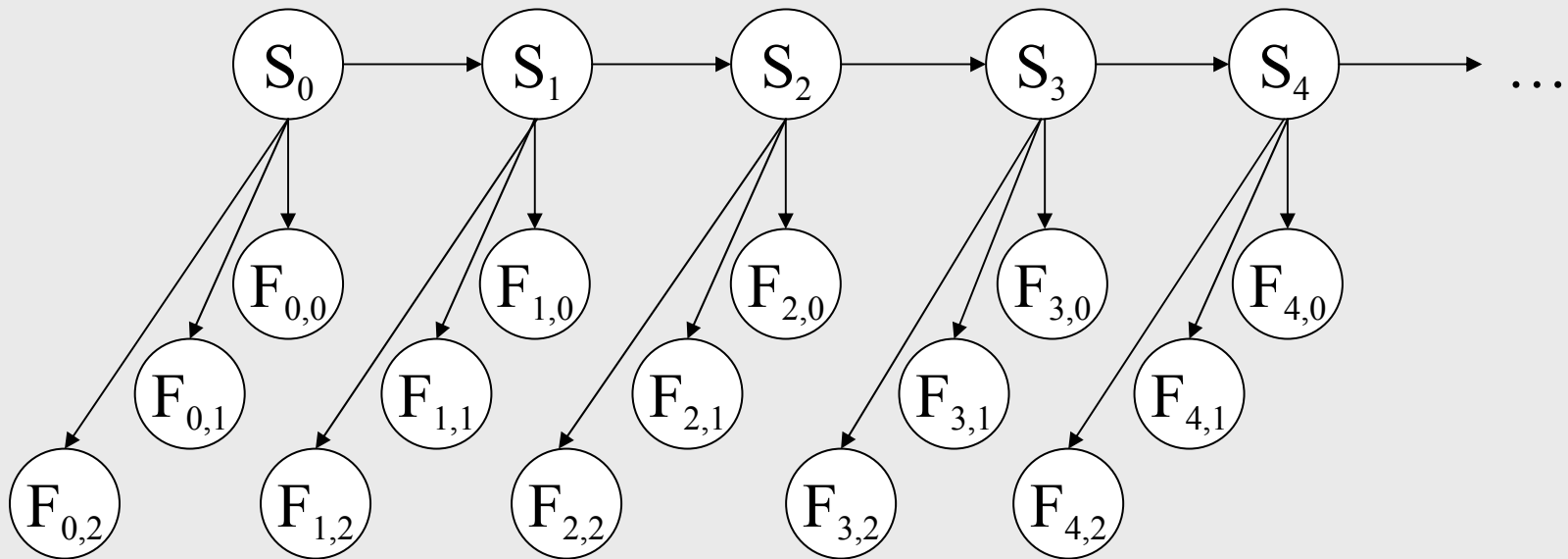
- Applications need richer input representations: multiple *overlapping* features, whole chunks of text

<i>Word features</i>	<i>Line features</i>
word identity	centered
capitalization	indentation
ends in “-tion”	white space ratio
word in word list	begins with number
word font	ends with “?”

- Generative models do not handle easily overlapping, non-independent features
- Alternative: *conditional* model $P(\mathbf{s}|\mathbf{o})$

Generating Multiple Features

- Instances: sequences of feature sets
 - Word identity
 - Word properties (eg. spelling, capitalization)
- Labelings: state sequences



- Removing conditional independence of features on states brings intractability

First Attempt: Maximum Entropy Markov Models

- Per-state conditional model



- Train each state model *separately* from labeled sequences
- Decoding: Viterbi
- Special case: Ratnaparkhi's POS tagger

Features

- Model $P(s|s',o)$ with *observation predicates*
 - o is the word “apple”
 - o is capitalized
 - o is on a left-justified line
- Each feature f conjoins
 - Observation predicate p
 - Source state s' and destination state s

$$f(t',o,t) = \begin{cases} 1 & \text{if } p(o) \text{ and } t' = s' \text{ and } t = s \\ 0 & \text{otherwise} \end{cases}$$

Application: Q-A pairs from FAQ

X-NNTP-Poster: NewsHound v1.33

Archive-name: acorn/faq/part2

Frequency: monthly

2.6) What configuration of serial cable should I use?

Here follows a diagram of the necessary connections for common terminal programs to work properly. They are as far as I know the informal standard agreed upon by commercial comms software developers for the Arc.

Pins 1, 4, and 8 must be connected together inside the 9 pin plug. This is to avoid the well known serial port chip bugs. The modem's DCD (Data Carrier Detect) signal has been re-routed to the Arc's RI (Ring Indicator) most modems broadcast a software RING signal anyway, and even then it's really necessary to detect it for the model to answer the call.

2.7) The sound from the speaker port seems quite muffled.

How can I get unfiltered sound from an Acorn machine?

All Acorn machine are equipped with a sound filter designed to remove high frequency harmonics from the sound output. To bypass the filter, hook into the Unfiltered port. You need to have a capacitor. Look for LM324 (chip 39) and and hook the capacitor like this:

Experimental Data

- 38 files belonging to 7 UseNet FAQs

```
<head>                X-NNTP-Poster: NewsHound v1.33
<head>                Archive-name: acom/faq/part2
<head>                Frequency: monthly
<head>
<question> 2.6) What configuration of serial cable should I use?
<answer>
<answer>                Here follows a diagram of the necessary connection
<answer>                programs to work properly. They are as far as I know
<answer>                agreed upon by commercial comms software developers fo
<answer>
<answer>                Pins 1, 4, and 8 must be connected together inside
<answer>                is to avoid the well known serial port chip bugs. The
```

- Procedure: for each FAQ, train on one file, test on others; average.

Features in Experiments

begins-with-number

begins-with-ordinal

begins-with-punctuation

begins-with-question-word

begins-with-subject

blank

contains-alphanum

contains-bracketed-number

contains-http

contains-non-space

contains-number

contains-pipe

contains-question-mark

contains-question-word

ends-with-question-mark

first-alpha-is-capitalized

indented

indented-1-to-4

indented-5-to-10

more-than-one-third-space

only-punctuation

prev-is-blank

prev-begins-with-ordinal

shorter-than-30

Models Tested

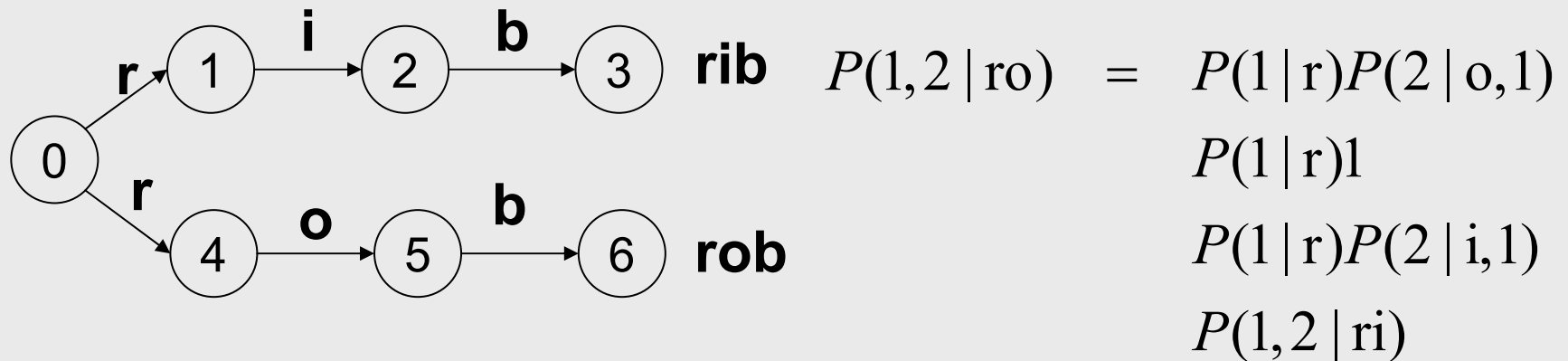
- **ME-Stateless**: A single maximum entropy classifier applied to each line independently.
- **TokenHMM**: A fully-connected HMM with four states, one for each of the line categories, each of which generates individual tokens (groups of alphanumeric characters and individual punctuation characters).
- **FeatureHMM**: Identical to TokenHMM, only the lines in a document are first converted to sequences of features.
- **MEMM**: maximum entropy Markov model

Results

<i>Learner</i>	<i>Segmentation precision</i>	<i>Segmentation recall</i>
ME-Stateless	0.038	0.362
TokenHMM	0.276	0.140
FeatureHMM	0.413	0.529
MEMM	0.867	0.681

Label Bias Problem

- Example (after Bottou '91):



- Bias toward states with fewer outgoing transitions.
- Per-state normalization does not allow the required $\text{score}(1, 2|ro) \ll \text{score}(1, 2|ri)$.

Proposed Solutions

- *Determinization*:

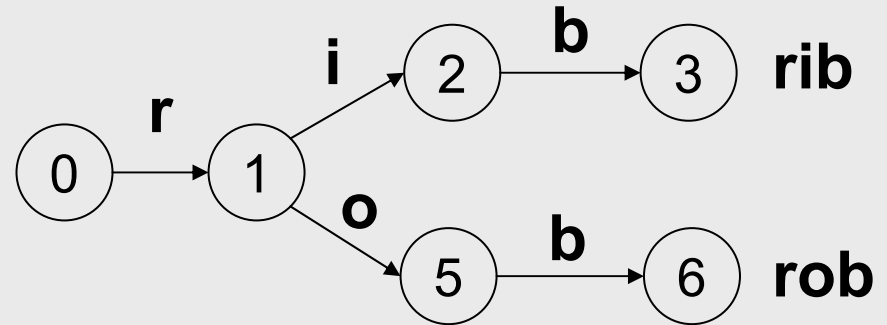
- Not always possible
- State-space explosion

- *Fully-connected* models:

- Lacks prior structural knowledge
- Problem remains for low-entropy next-state distributions

- *Conditional random fields* (CRFs):

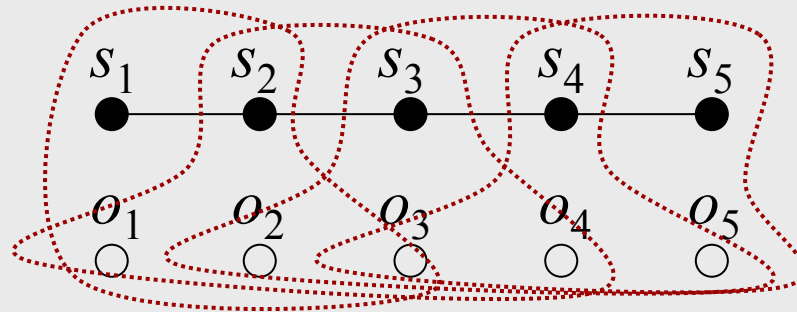
- Allow some transitions to *vote* more strongly than others
- *Whole sequence* rather than per-state normalization



Conditional Undirected Models

- $P(\text{state sequence } \mathbf{s} \mid \text{observation sequence } \mathbf{o})$
instead of $P(\mathbf{o}, \mathbf{s})$
- Allow arbitrary dependencies on \mathbf{o}
- Efficient inference is dependencies within \mathbf{s} are constrained
- States don't need to encode dependency on past and future observations

Conditional Random Fields



- Markov on \mathbf{s} , conditional dependency on \mathbf{o}

$$P_{\Lambda}(\mathbf{s} \mid \mathbf{o}) = \frac{1}{Z_{\Lambda}(\mathbf{o})} \prod_i \exp \Lambda \cdot \mathbf{f}_i(s_{i-1}, s_i, \mathbf{o})$$

- Feature vectors \mathbf{f}_i represent interactions between successive states, distribution of individual states

From HMMs to CRFs

$$\mathbf{s} = s_1 \dots s_n \quad \mathbf{o} = o_1 \dots o_n$$

HMM
$$P(\mathbf{s} | \mathbf{o}) = \frac{P(s_0)}{P(\mathbf{o})} \prod_{t=1}^n P(s_t | s_{t-1}) P(o_t | s_t)$$

MEMM
$$P(\mathbf{s} | \mathbf{o}) = \prod_{t=1}^n \frac{1}{Z(s_{t-1}, o_t)} \exp \left(\begin{array}{l} \Lambda \cdot \mathbf{f}(s_t, s_{t-1}) + \\ \Omega \cdot \mathbf{g}(s_t, o_t) \end{array} \right)$$

CRF
$$P(\mathbf{s} | \mathbf{o}) = \frac{1}{Z(\mathbf{o})} \prod_{t=1}^n \exp \left(\begin{array}{l} \Lambda \cdot \mathbf{f}(s_t, s_{t-1}) + \\ \Omega \cdot \mathbf{g}(s_t, o_t) \end{array} \right)$$

Decoding

- Viterbi algorithm computes

$$\hat{\mathbf{s}} = \arg \max_{\mathbf{s}} P_{\Lambda}(\mathbf{s} \mid \mathbf{o}) = \arg \max_{\mathbf{s}} \Lambda \cdot \Phi(\mathbf{o}, \mathbf{s})$$

$$\Phi(\mathbf{o}, \mathbf{s}) = \sum_i \mathbf{f}_i(s_{i-1}, s_i, \mathbf{o})$$

- *Linear sequence model*: clean separation between
 - Decoding
 - Training
 - As in linear classification

Efficient Estimation

- Matrix notation

$$M_t(s', s | \mathbf{o}) = \exp \Lambda \cdot \mathbf{f}_i(s_{t-1}, s_t, \mathbf{o})$$

$$P_\Lambda(\mathbf{s} | \mathbf{o}) = \frac{1}{Z_\Lambda(\mathbf{o})} \prod_t M_i(s_{t-1}, s_t | \mathbf{o})$$

$$Z_\Lambda(\mathbf{o}) = (M_1(\mathbf{o})M_2(\mathbf{o})\mathbf{L} \ M_{n+1}(\mathbf{o}))_{\text{start,stop}}$$

- Efficient normalization: *forward-backward* algorithm

Forward-Backward Calculations

- For any *path function* $G(s) = \sum_t g_t(s_{t-1}, s_t)$

$$\begin{aligned} E_{\mathbf{S} \sim P(\mathbf{S} | \mathbf{o})} G(\mathbf{S}) &= \sum_{\mathbf{s}} P_{\Lambda}(\mathbf{s} | \mathbf{o}) G(\mathbf{s}) \\ &= \sum_t \frac{\alpha_t(\mathbf{o}) [g_{t+1} * M_{t+1}(\mathbf{o})] \beta'_{t+1}(\mathbf{o})}{Z_{\Lambda}(\mathbf{o})} \end{aligned}$$

$$\alpha_t(\mathbf{o}) = \alpha_{t-1}(\mathbf{o}) M_t(\mathbf{o})$$

$$\beta'_t(\mathbf{o}) = M_{t+1}(\mathbf{o}) \beta'_{t+1}(\mathbf{o})$$

$$Z_{\Lambda}(\mathbf{o}) = \alpha_{n+1}(\text{end} | \mathbf{o}) = \beta_0(\text{start} | \mathbf{o})$$

- Easy generalization to constrained set of paths

Training

- Maximize $L(\Lambda) = \sum_k \log P_\Lambda(\mathbf{s}_k | \mathbf{o}_k)$
- Log-likelihood *gradient*

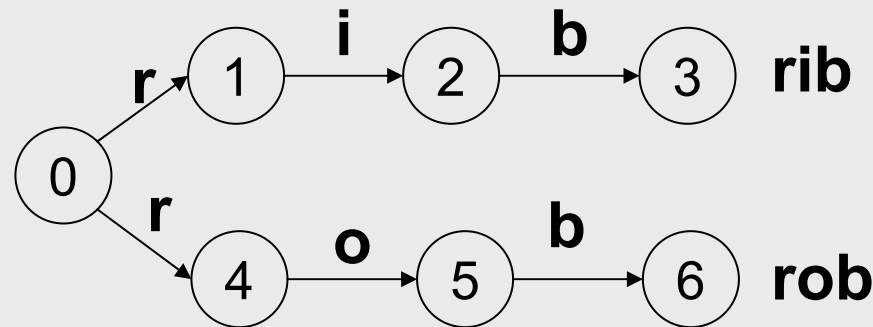
$$\nabla L(\Lambda) = \sum_k \left(\Phi(\mathbf{o}_k, \mathbf{s}_k) - E_{\mathbf{S} \sim P_\Lambda(\mathbf{S} | \mathbf{o}_k)} \Phi(\mathbf{o}_k, \mathbf{S}) \right)$$

- Methods: iterative scaling, *conjugate gradient*
- *Partially-observable* case (labeled states)

$$\nabla L(\Lambda) = \sum_k \begin{pmatrix} E_{\mathbf{S} \sim P_\Lambda(\mathbf{S} | \mathbf{o}_k, \mathbf{I}(\mathbf{S}) = \mathbf{I}_k)} \Phi(\mathbf{o}_k, \mathbf{S}) \\ - E_{\mathbf{S} \sim P_\Lambda(\mathbf{S} | \mathbf{o}_k)} \Phi(\mathbf{o}_k, \mathbf{S}) \end{pmatrix}$$

Label Bias Experiment

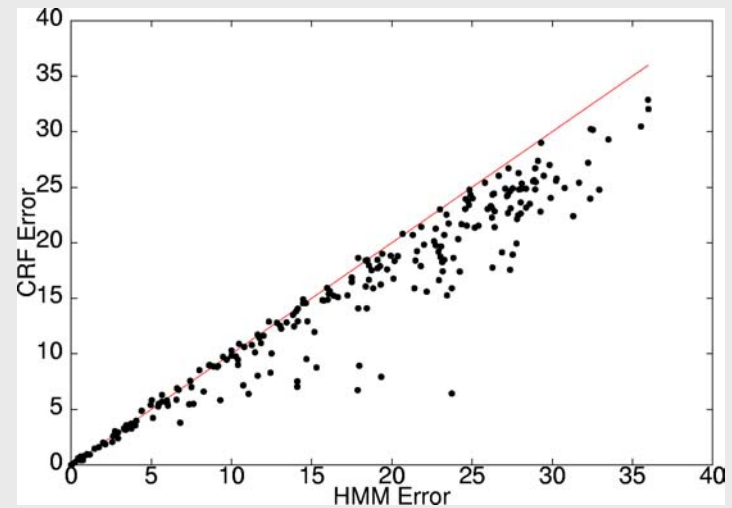
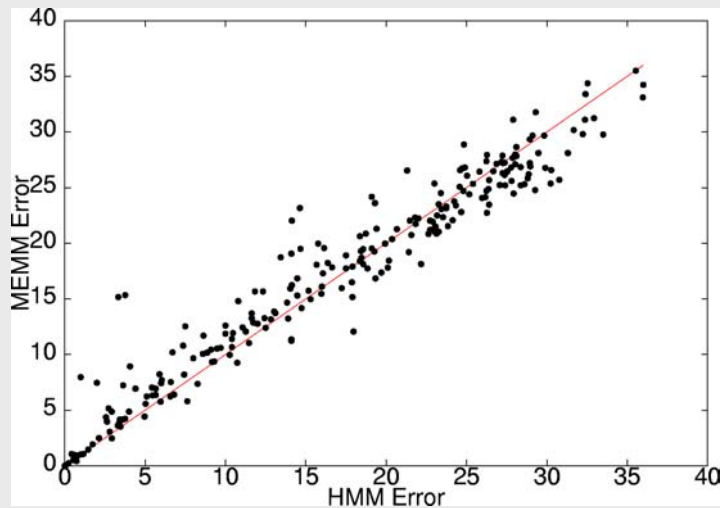
- Data source: noisy version of



- $P(\text{intended symbol}) = 29/30$, $P(\text{other}) = 1/30$.
- Train both an MEMM and a CRF with identical topologies on data from the source.
- Compute decoding error: CRF 4.6%, MEMM 42% (2,000 training samples, 500 test)

Mixed-Order Sources

- Data generated by mixing sparse first and second order HMMs with varying mixing coefficient.
- Modeled by first-order HMM, MEMM and CRF (without contextual or overlapping features).



Part-of-Speech Tagging

- Trained on 50% of the 1.1 million words in the Penn treebank. In this set, 5.45% of the words occur only once, and were mapped to “oov”.
- Experiments with two different sets of features:
 - traditional: just the words
 - take advantage of power of conditional models: use words, plus overlapping features: capitalized, begins with #, contains hyphen, ends in -ing, -ogy, -ed, -s, -ly, -ion, -tion, -ity, -ies.

POS Tagging Results

<i>model</i>	<i>error</i>	<i>oov error</i>
HMM	5.69%	45.99%
MEMM	6.37%	54.61%
CRF	5.55%	48.05%
MEMM+	4.81%	26.99%
CRF+	4.27%	23.76%

Shallow Parsing

- Noun phrase chunking from POS-tagged text

Rockwell International Corp. 's Tulsa unit said it signed a tentative agreement extending its contract with Boeing Co. to provide structural parts for Boeing 's 747 jetliners

- Standard benchmark task

NP Chunking Results

	F
<i>CRF</i>	94.39%
SVM combination [Kudo & Matsumoto 01]	94.22%
Winnow [Zhang, Damerau, & Johnson 02]	93.89%
<i>Voted perceptron</i> [Collins 02]	93.53%

$$F = \frac{2PR}{P + R}$$

- Warning: different feature sets

Training Tricks

- Preconditioned conjugate gradient
 - Approximate diagonal of Hessian (exact diagonal is too expensive to compute)
- Features
 - Input predicate & transition predicate
 - Pre-compute input predicates
- Full forward-backward
 - Pruned forward-backward may be needed for larger models

Alternative Training Methods

- Generalized perceptron [Collins 02]

$$\mathbf{w}_0 = \mathbf{0}; k = 0$$

for $t = 1, \dots, T$

for $i = 1, \dots, N$

$$\hat{\mathbf{s}} = \arg \max_{\mathbf{s}} \mathbf{w}_k \cdot \Phi(\mathbf{o}_i, \mathbf{s})$$

$$\text{if } \hat{\mathbf{s}} \neq \mathbf{s}_i \text{ then } \mathbf{w}_{k+1} = \mathbf{w}_k + \Phi(\mathbf{o}_i, \mathbf{s}_i) - \Phi(\mathbf{o}_i, \hat{\mathbf{s}}) \text{ else } \mathbf{w}_{k+1} = \mathbf{w}_k$$

$$k \leftarrow k + 1$$

$$\Lambda = \sum_k \mathbf{w}_k / NT$$

- Questions:

- Convergence speed
- Does it lose something from Viterbi?

Next Steps

- Compare with other linear sequence models
 - Generalized perceptron [Collins 02]
 - Margin and generalization bounds?
- Parsing
 - Trees instead of chains
 - Inside-outside replaces forward-backward
 - Computationally challenging: very large label set
- General graphs
 - Suggestive results for *collective classification* [Taskar & al 02]