

## 7 Sequential Models

1. Sequential Inference Problems
2. HMM
3. Advanced Models

# Tutorial on Machine Learning in Natural Language Processing and Information Extraction

Dan Roth

University of Illinois, Urbana-Champaign

[danr@cs.uiuc.edu](mailto:danr@cs.uiuc.edu)

<http://L2R.cs.uiuc.edu/~danr>

# Outline

---

- Shallow Parsing
  - What it is
  - Why we need it
- Shallow Parsing (Learning) Models
  - Learning Sequences
- Hidden Markov Model (HMM)
- Discriminative Approaches
  - HMM with Classifiers
  - PMM
- Learning and Inference
  - CSCL
- Related Approaches

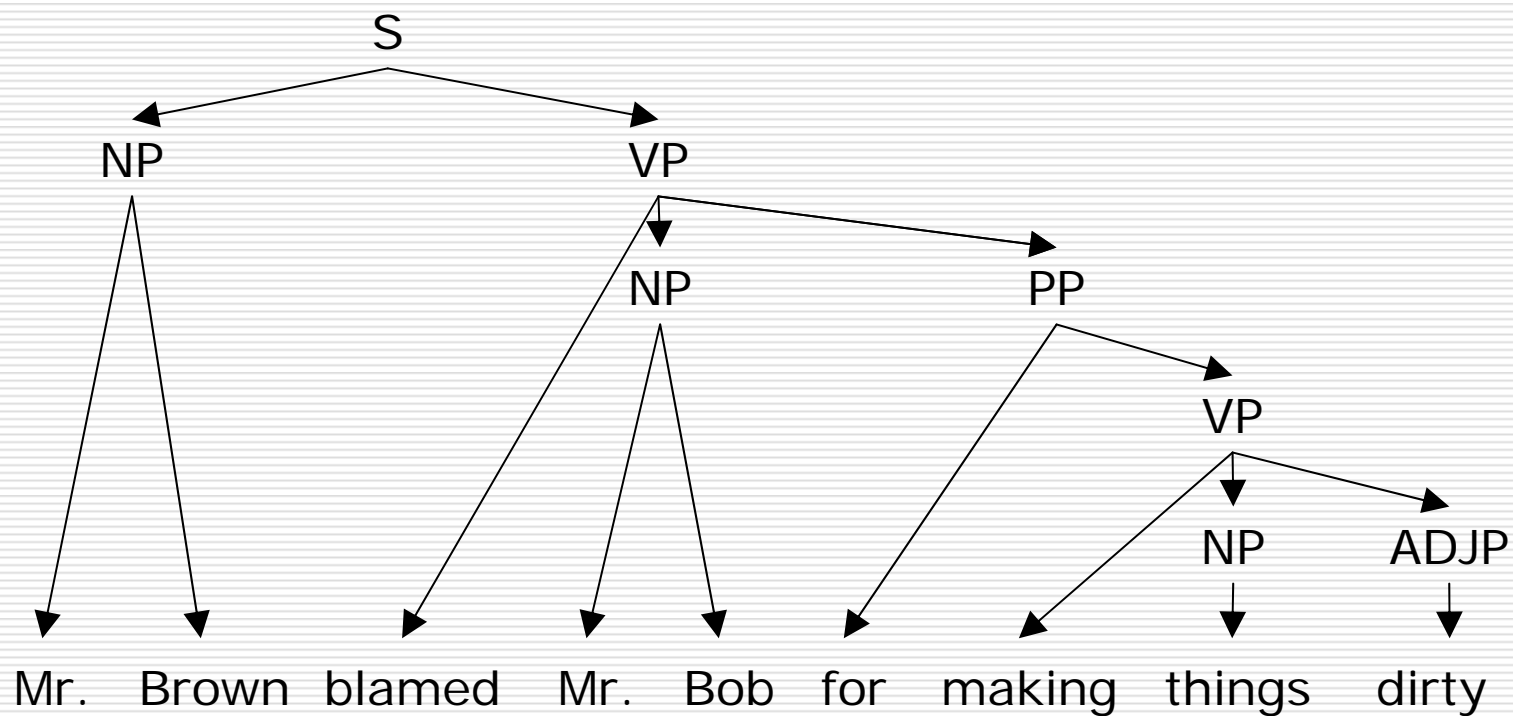
# Parsing

---

- Parsing is a task to analyze for syntactical structure of inputs
- An output is a “parse tree”

# Parsing

---



# Parsing

---

- Parsing is an important task toward understanding natural languages
- It is not an easy task
- Many NLP tasks do not require all information from parse trees

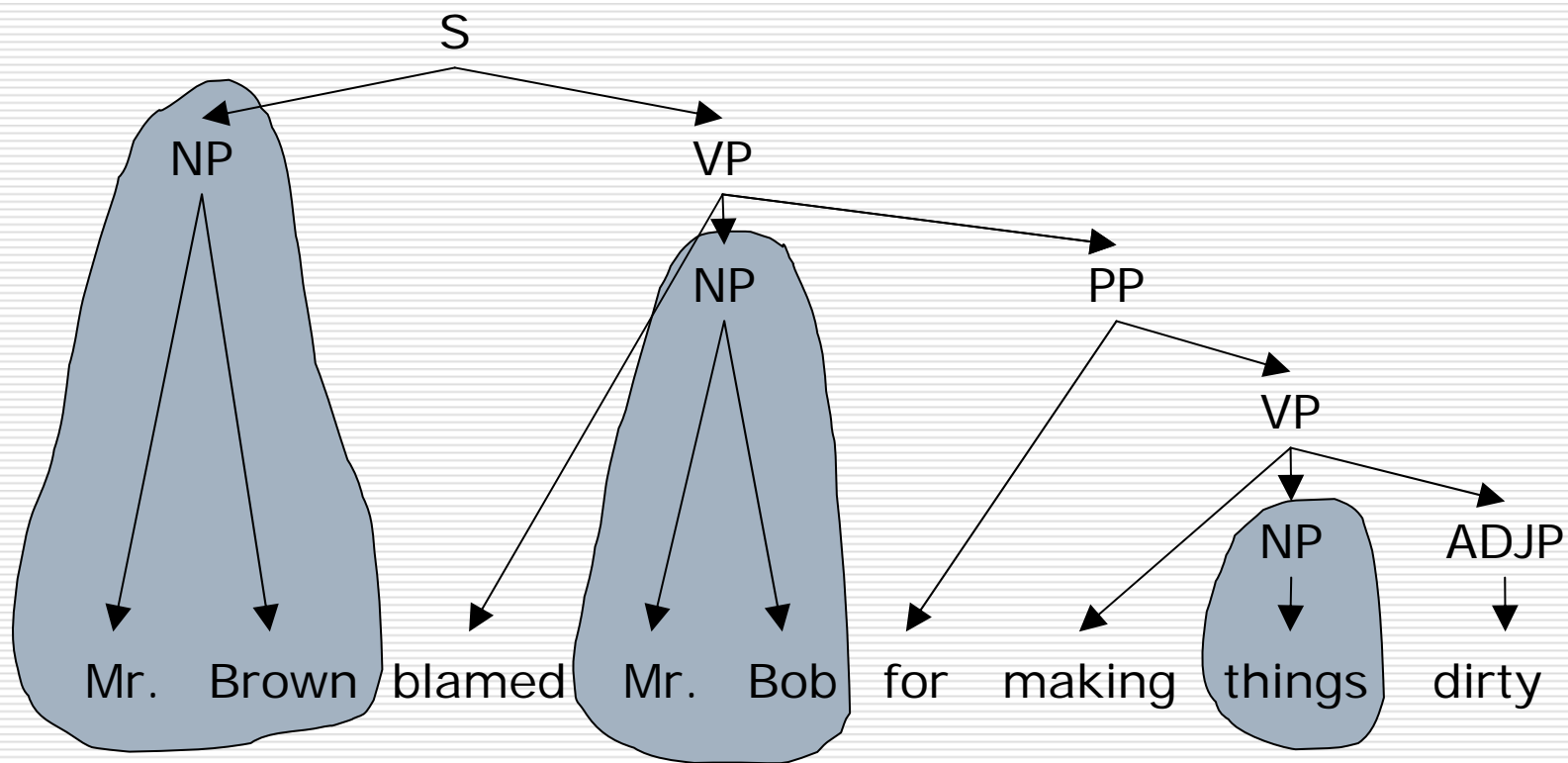
# Shallow Parsing

---

- Shallow Parsing is a generic term for a task identifies only a subset of parse trees

# Shallow Parsing

---



# Reasons for Studying Shallow Parsing

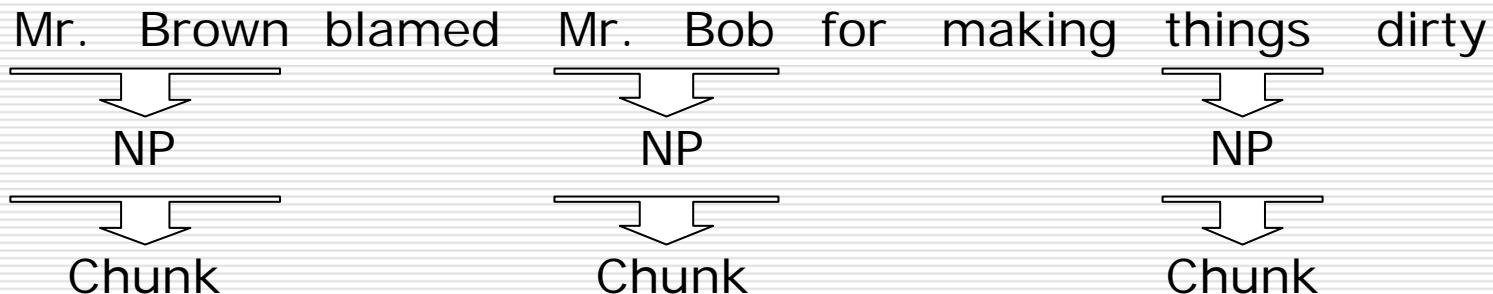
---

- Not all the parse tree information is needed
  - It should do better than full parsing in those specific parts
- It can be done more robustly
  - More adaptive to data from new domains (Li & Roth' CoNLL'01)
- ~~□ Shallow parsing as an intermediate step toward full parsing~~
- It is a generic chunking task – applicable to many other segmentation tasks such as named entity recognition. [[LBJ generic segmentation implementation](#)]

# Shallow Parsing [Today]

---

- By shallow parsing we mean: identifying non-overlapping, non-embedding phrases.



- Shallow Parsing = Text Chunking

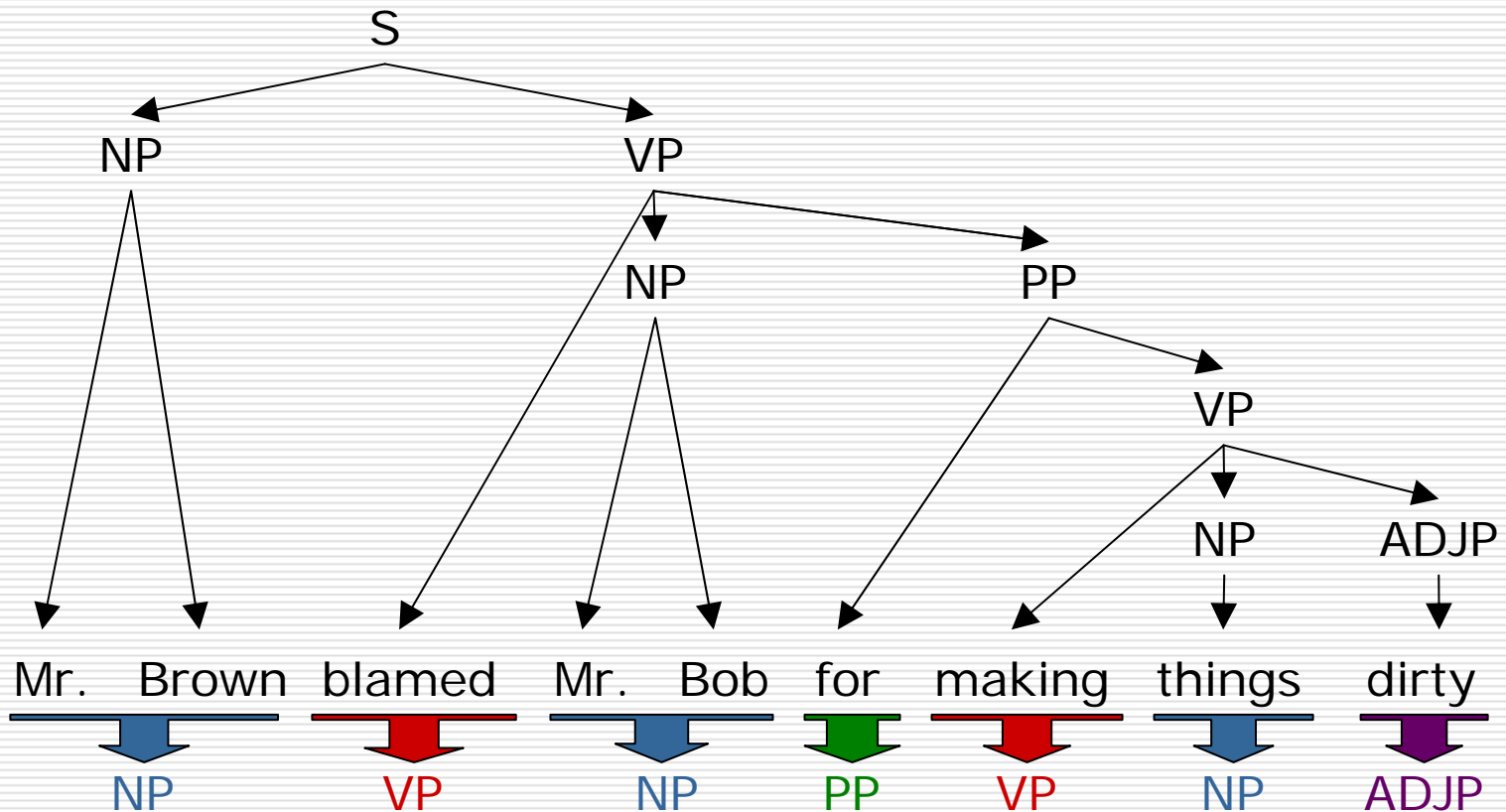
# Text Chunking

---

- Usually text chunking without any further specification follows the definition from CoNLL-2000 shared task [[Demo](#)]

# Text Chunking

---



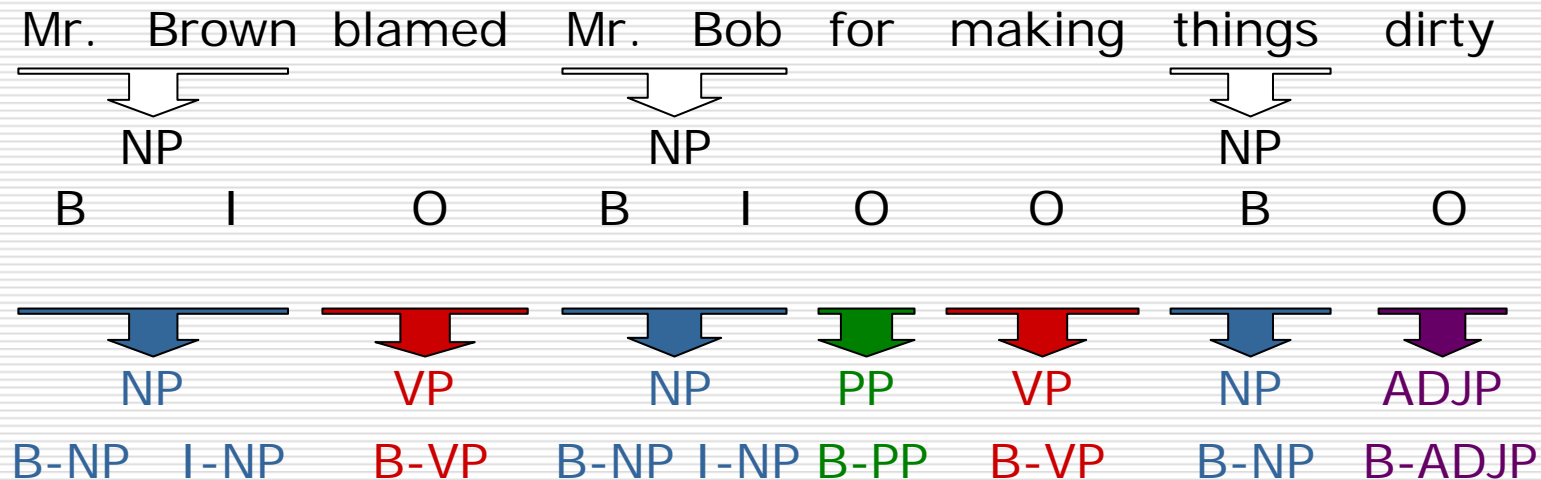
# Modeling Shallow Parsing

---

- Model shallow parsing as a sequence prediction problem
- For each word predict one of these :
  - B – Beginning of a chunk
  - I – Inside a chunk but not a beginning
  - O – Outside a chunk

# Modeling Shallow Parsing

---



# Similar Problems

---

- This model applies to many other problems
  - NLP
    - Named entity recognition
    - Verb-argument identification
  - Other domains
    - Identifying genes (splice sites, Chuang&Roth'01)

# Outline

---

- Shallow Parsing
    - What it is
    - Why we need it
  - Shallow Parsing (Learning) Models
    - Learning Sequences
  - Hidden Markov Model (HMM)
  - Discriminative Approaches
    - HMM with Classifiers
    - PMM
  - Learning and Inference
    - CSCL
  - Related Approaches
- 

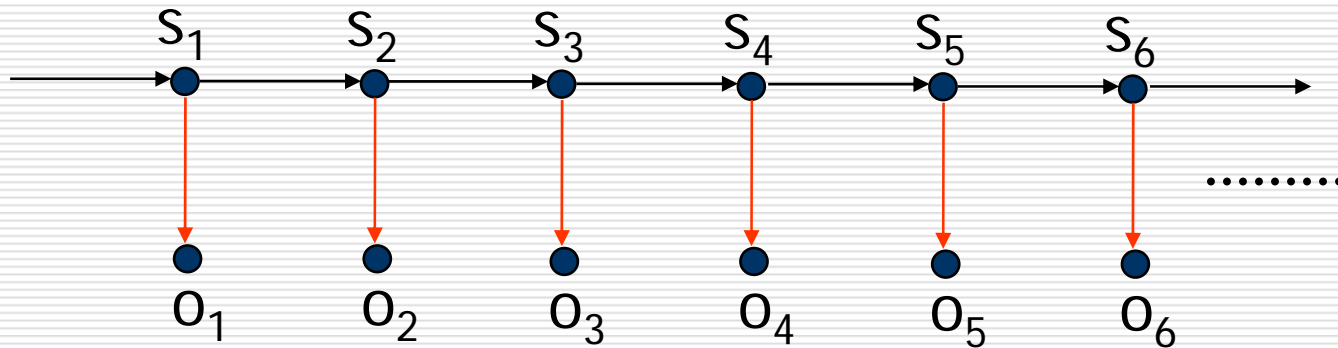
# Hidden Markov Model (HMM)

---

- HMM is a probabilistic generative model
  - It models how an observed sequence is generated
- Let's call each position in a sequence a time step
- At each time step, there are two variables
  - Current state (hidden)
  - Observation

# HMM

---



## □ Elements

- Initial state probability  $P(s_1)$
- Transition probability  $P(s_t | s_{t-1})$
- Observation probability  $P(o_t | s_t)$

# HMM for Shallow Parsing

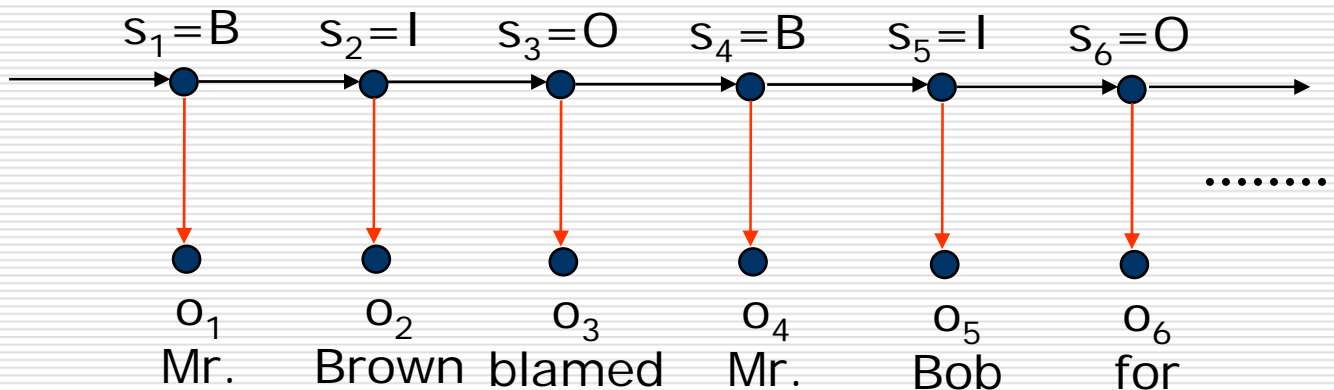
---

- States:

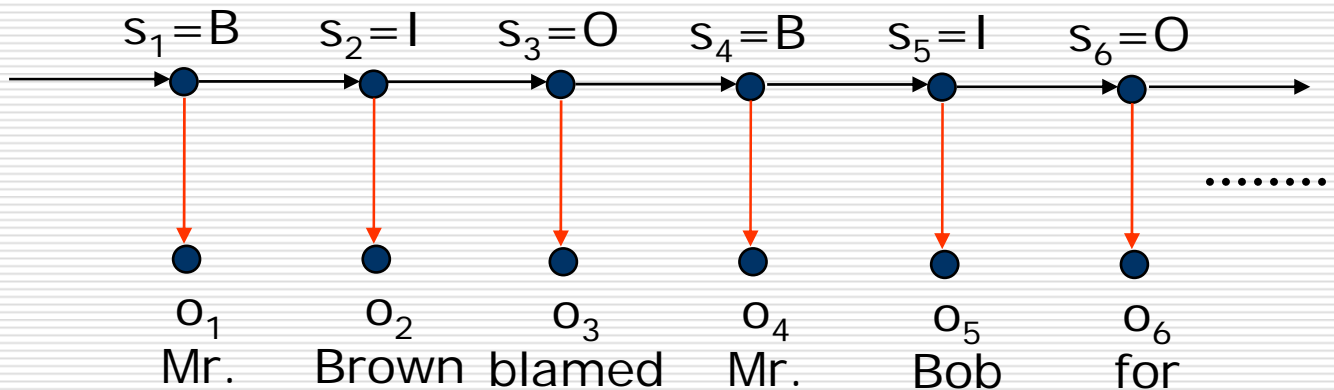
- {B, I, O}

- Observations:

- Actual words and/or part-of-speech tags



# HMM for Shallow Parsing



Initial state probability:  $P(s_1=B)$   
 Transition probability:  $P(s_t=B | s_{t-1}=I), P(s_t=I | s_{t-1}=O), P(s_t=O | s_{t-1}=B), \dots$   
 Observation Probability:  $P(o_t='Mr.' | s_t=B), P(o_t='Brown' | s_t=I), P(o_t='blamed' | s_t=O), \dots$

- Given a sentence, we can ask what the most likely state sequence is

# Finding most likely state sequence in HMM (1)

---

$$\begin{aligned} & P(s_k, s_{k-1}, \dots, s_1, o_k, o_{k-1}, \dots, o_1) \\ &= P(o_k | o_{k-1}, o_{k-2}, \dots, o_1, s_k, s_{k-1}, \dots, s_1) \\ & \quad \cdot P(o_{k-1}, o_{k-2}, \dots, o_1, s_k, s_{k-1}, \dots, s_1) \\ &= P(o_k | s_k) \cdot P(o_{k-1}, o_{k-2}, \dots, o_1, s_k, s_{k-1}, \dots, s_1) \\ &= P(o_k | s_k) \cdot P(s_k | s_{k-1}, s_{k-2}, \dots, s_1, o_{k-1}, o_{k-2}, \dots, o_1) \\ & \quad \cdot P(s_{k-1}, s_{k-2}, \dots, s_1, o_{k-1}, o_{k-2}, \dots, o_1) \\ &= P(o_k | s_k) \cdot P(s_k | s_{k-1}) \\ & \quad \cdot P(s_{k-1}, s_{k-2}, \dots, s_1, o_{k-1}, o_{k-2}, \dots, o_1) \\ &= P(o_k | s_k) \cdot \left[ \prod_{t=1}^{k-1} P(s_{t+1} | s_t) \cdot P(o_t | s_t) \right] \cdot P(s_1) \end{aligned}$$

# Finding most likely state sequence in HMM (2)

$$\arg \max_{s_k, s_{k-1}, \dots, s_1} P(s_k, s_{k-1}, \dots, s_1 | o_k, o_{k-1}, \dots, o_1)$$

$$= \arg \max_{s_k, s_{k-1}, \dots, s_1} \frac{P(s_k, s_{k-1}, \dots, s_1, o_k, o_{k-1}, \dots, o_1)}{P(o_k, o_{k-1}, \dots, o_1)}$$

$$= \arg \max_{s_k, s_{k-1}, \dots, s_1} P(s_k, s_{k-1}, \dots, s_1, o_k, o_{k-1}, \dots, o_1)$$

$$= \arg \max_{s_k, s_{k-1}, \dots, s_1} P(o_k | s_k) \cdot \left[ \prod_{t=1}^{k-1} P(s_{t+1} | s_t) \cdot P(o_t | s_t) \right] \cdot P(s_1)$$

# Finding most likely state sequence in HMM (3)

$$\begin{aligned}
 & \max_{s_k, s_{k-1}, \dots, s_1} P(o_k | s_k) \cdot \left[ \prod_{t=1}^{k-1} P(s_{t+1} | s_t) \cdot P(o_t | s_t) \right] \cdot P(s_1) \\
 = & \max_{s_k} P(o_k | s_k) \cdot \max_{s_{k-1}, \dots, s_1} \left[ \prod_{t=1}^{k-1} P(s_{t+1} | s_t) \cdot P(o_t | s_t) \right] \cdot P(s_1) \\
 = & \max_{s_k} P(o_k | s_k) \cdot \max_{s_{k-1}} [P(s_k | s_{k-1}) \cdot P(o_{k-1} | s_{k-1})] \\
 & \cdot \max_{s_{k-2}, \dots, s_1} \left[ \prod_{t=1}^{k-2} P(s_{t+1} | s_t) \cdot P(o_t | s_t) \right] \cdot P(s_1) \\
 = & \max_{s_k} P(o_k | s_k) \cdot \max_{s_{k-1}} [P(s_k | s_{k-1}) \cdot P(o_{k-1} | s_{k-1})] \\
 & \cdot \max_{s_{k-2}} [P(s_{k-1} | s_{k-2}) \cdot P(o_{k-2} | s_{k-2})] \cdot \dots \\
 & \cdot \max_{s_1} [P(s_2 | s_1) \cdot P(o_1 | s_1)] \cdot P(s_1)
 \end{aligned}$$

# Finding most likely state sequence in HMM (4)

---

$$\begin{aligned} & \max_{s_k} P(o_k | s_k) \cdot \max_{s_{k-1}} [P(s_k | s_{k-1}) \cdot P(o_{k-1} | s_{k-1})] \\ & \cdot \max_{s_{k-2}} [P(s_{k-1} | s_{k-2}) \cdot P(o_{k-2} | s_{k-2})] \cdot \dots \\ & \cdot \max_{s_2} [P(s_3 | s_2) \cdot P(o_2 | s_2)] \cdot \\ & \cdot \max_{s_1} [P(s_2 | s_1) \cdot P(o_1 | s_1)] \cdot P(s_1) \end{aligned}$$

- Viterbi's Algorithm
  - Dynamic Programming

# Learning the Model

---

## □ Estimate

- Initial state probability  $P(s_1)$
- Transition probability  $P(s_t | s_{t-1})$
- Observation probability  $P(o_t | s_t)$

## □ Unsupervised Learning

- EM Algorithm

## □ Supervised Learning

- Estimate each element directly from data

# Experimental Results

---

## □ NP Prediction (POS Tags Only)

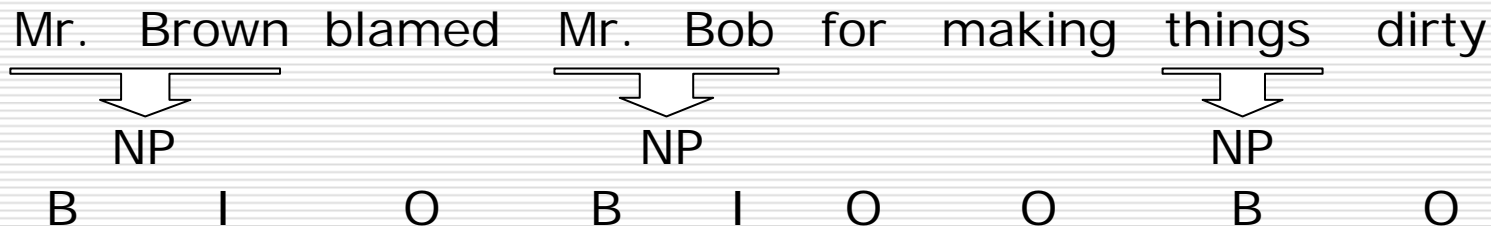
- 89.08 Recall
- 86.62 Precision
- 87.83 F1

$$R(\text{Recall}) = \frac{\text{Number of chunks predicted correctly}}{\text{Number of correct chunks}}$$
$$P(\text{Precision}) = \frac{\text{Number of chunks predicted correctly}}{\text{Number of predicted chunks}}$$
$$F_1 = \frac{2RP}{R + P}$$

# Note

---

- Experiments use a different representation




[Mr. Brown] blamed [Mr. Bob] for making [things] dirty

# Outline

---

- Shallow Parsing
    - What it is
    - Why we need it
  - Shallow Parsing (Learning) Models
    - Learning Sequences
  - Hidden Markov Model (HMM)

---

  - Discriminative Approaches
    - HMM with Classifiers
    - PMM
  - Learning and Inference
    - CSCL
  - Related Approaches
- 

# Problems with HMM

---

- Long-term dependencies are hard to incorporate
- HMM is trained to maximize the likelihood of the data, not to maximize the predictions
  - Maximize  $P(S,O)$ , not  $P(S|O)$
  - What metric do we care about?

# Discriminative Approaches

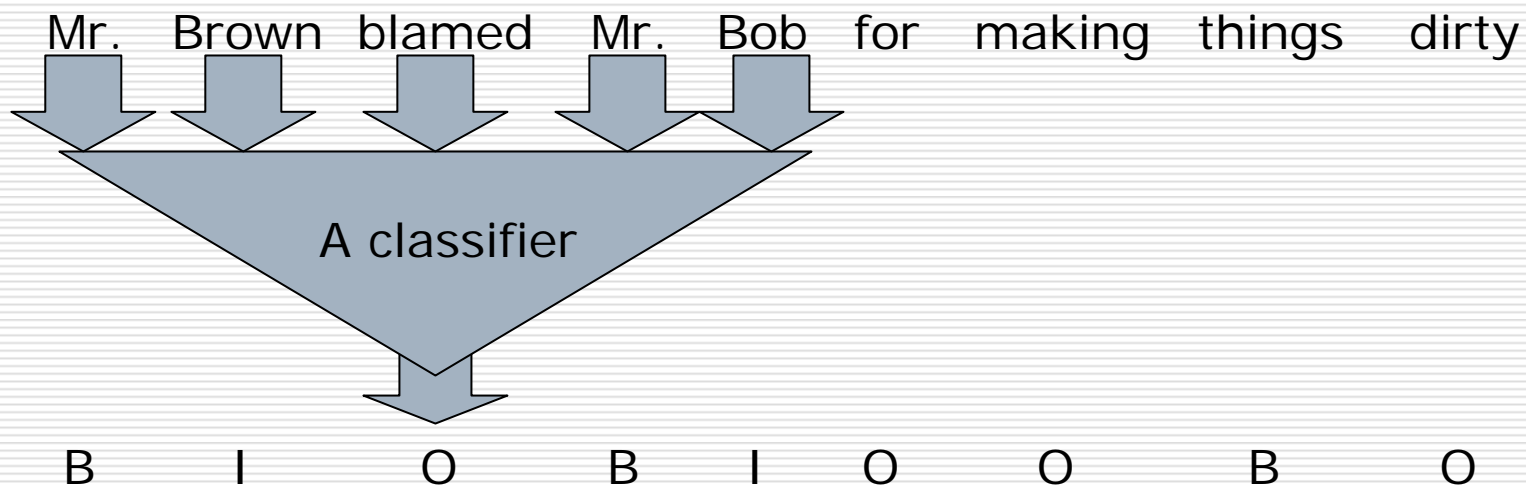
---

- Model the predictions directly
- Shallow Parsing
  - Goal: predict BIO sequences given a sentence
    - $S^* = \operatorname{argmax} P(S|O)$
  - Generative approaches model  $P(S,O)$
  - Discriminative approaches model  $P(S|O)$  directly

# Discriminative Approaches

---

- Use classifiers to predict the labels based on the context of the inputs

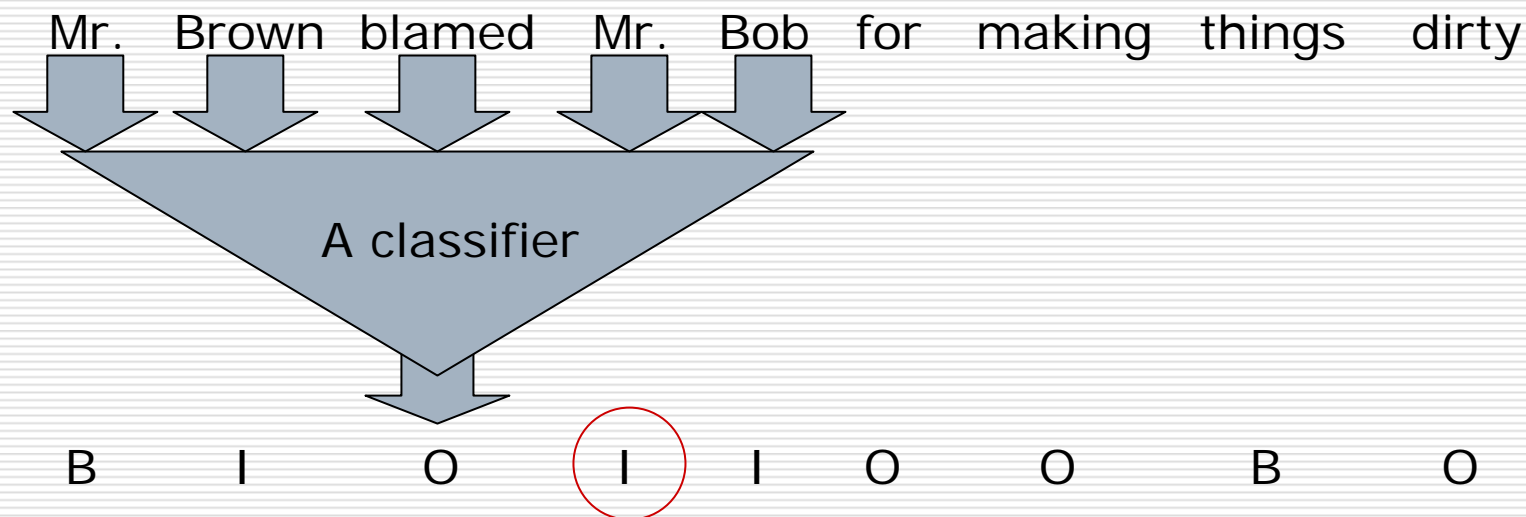


- Good: larger context can be incorporated

# Problems with using Classifiers

---

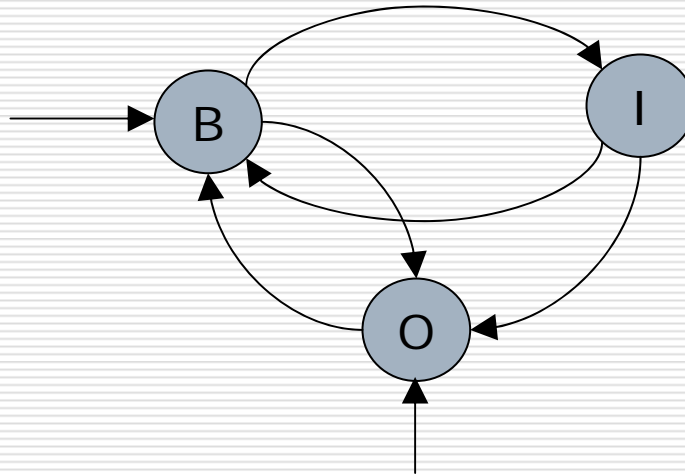
- Outputs may be inconsistent



# Constraints

---

- ❑ An 'I' cannot follow an 'O'
- ❑ The sequence has to begin with a 'B' or an 'O'



- ❑ How to maintain constraints?

# HMM revisited

---

- HMM does not have this problem.
- These constraints are taken care for by the transition probabilities.
- Specifically, zero transition probabilities ensure that outputs always satisfy constraints

# HMM with Classifiers

---

- HMM requires observation probability
  - $P(o_t|s_t)$
- Classifiers give us
  - $P(s_t|o_t)$
  
- We can compute  $P(o_t|s_t)$  by
  - $P(o_t|s_t) = P(s_t|o_t)P(o_t)/P(s_t)$
  - See details [here](#)

# Experimental Results

---

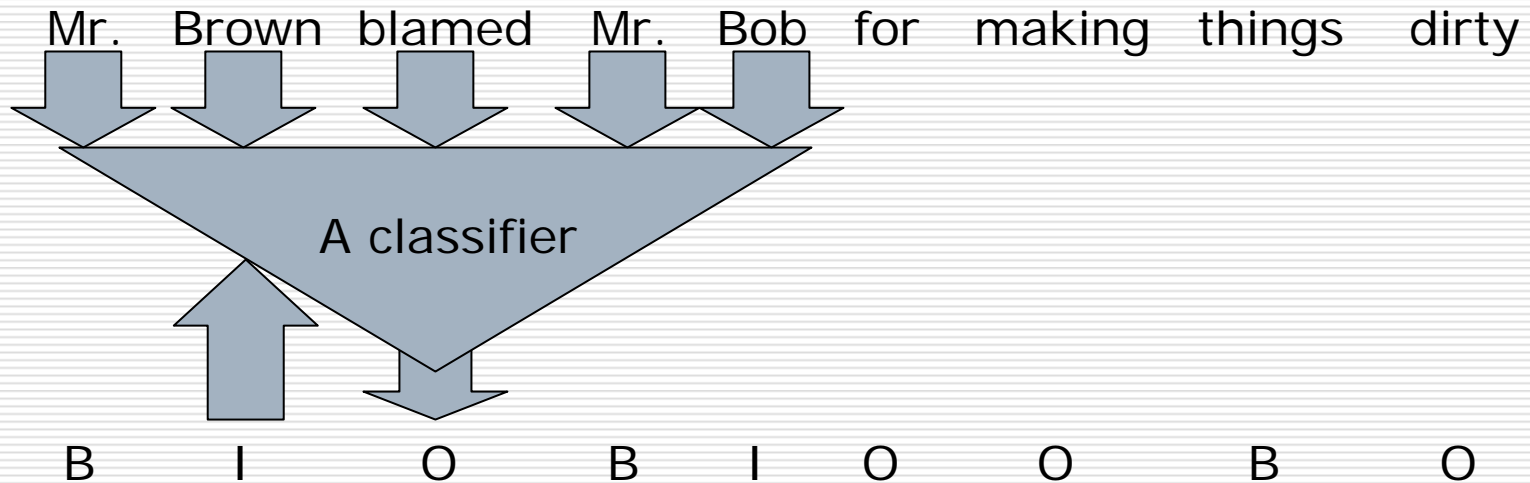
## □ NP Prediction

	Recall	Prediction	F1
HMM	89.08	86.62	87.83
HMM with Classifiers	91.49	91.53	91.51
HMM with Classifiers (with Lexical features)	93.85	93.46	93.65

# Projection based Markov Model (PMM)

---

- Classifiers use previous prediction as part of inputs



# PMM

---

- Each classifier outputs  $P(s_t | s_{t-1}, O)$

$$\arg \max_{s_k, s_{k-1}, \dots, s_1} P(s_k, s_{k-1}, \dots, s_1 | O)$$

$$= \arg \max_{s_k, s_{k-1}, \dots, s_1} P(s_k | s_{k-1}, \dots, s_1, O) \cdot P(s_{k-1}, \dots, s_1, O)$$

$$= \arg \max_{s_k, s_{k-1}, \dots, s_1} P(s_k | s_{k-1}, O) \cdot P(s_{k-1}, \dots, s_1, O)$$

$$= \arg \max_{s_k, s_{k-1}, \dots, s_1} \left[ \prod_{t=2}^k P(s_t | s_{t-1}, O) \right] \cdot P(s_1 | O)$$

- How do you Train/Test in this case?

# Experimental Results

---

## □ NP Prediction

	Recall	Precision	F1
HMM (POS)	89.08	86.62	87.83
HMM with Classifiers (POS)	91.49	91.53	91.51
HMM with Classifiers	93.85	93.46	93.65
PMM (POS)	92.04	91.77	91.90
PMM	94.28	94.02	94.15

# Outline

---

- Shallow Parsing
    - What it is
    - Why we need it
  - Shallow Parsing (Learning) Models
    - Learning Sequences
  - Hidden Markov Model (HMM)
  - Discriminative Approaches
    - HMM with Classifiers
    - PMM

---

  - Learning and Inference
    - CSCL
  - Related Approaches
- 

# Learning and Inference

---

- Learning: estimate parameters from data, and makes predictions for new data
- Inference: ensure that outputs satisfy constraints

# Inference

---

- Assign values to the variables of interest (output; states) in a way that maximizes your objective function and satisfies constraints

# Inference

---

## □ Boolean Constraint Satisfaction Problem

- $V$  – set of variables
- Cost –  $c: V \rightarrow [0,1]$
- Clauses – model constraints
- Satisfying assignment –  $\tau: V \rightarrow \{0,1\}$
- Find the solution  $\tau$  that *minimize the cost*

$$C(\tau) = \sum_{i=1..n} \tau(v_i)c(v_i)$$

# Inference for Shallow Parsing

---

- Define a variable  $v_i$  for each possible chunk
- Cost of each chunk =  $-P(v_i \text{ is a chunk})$
- For any two overlapping chunks:

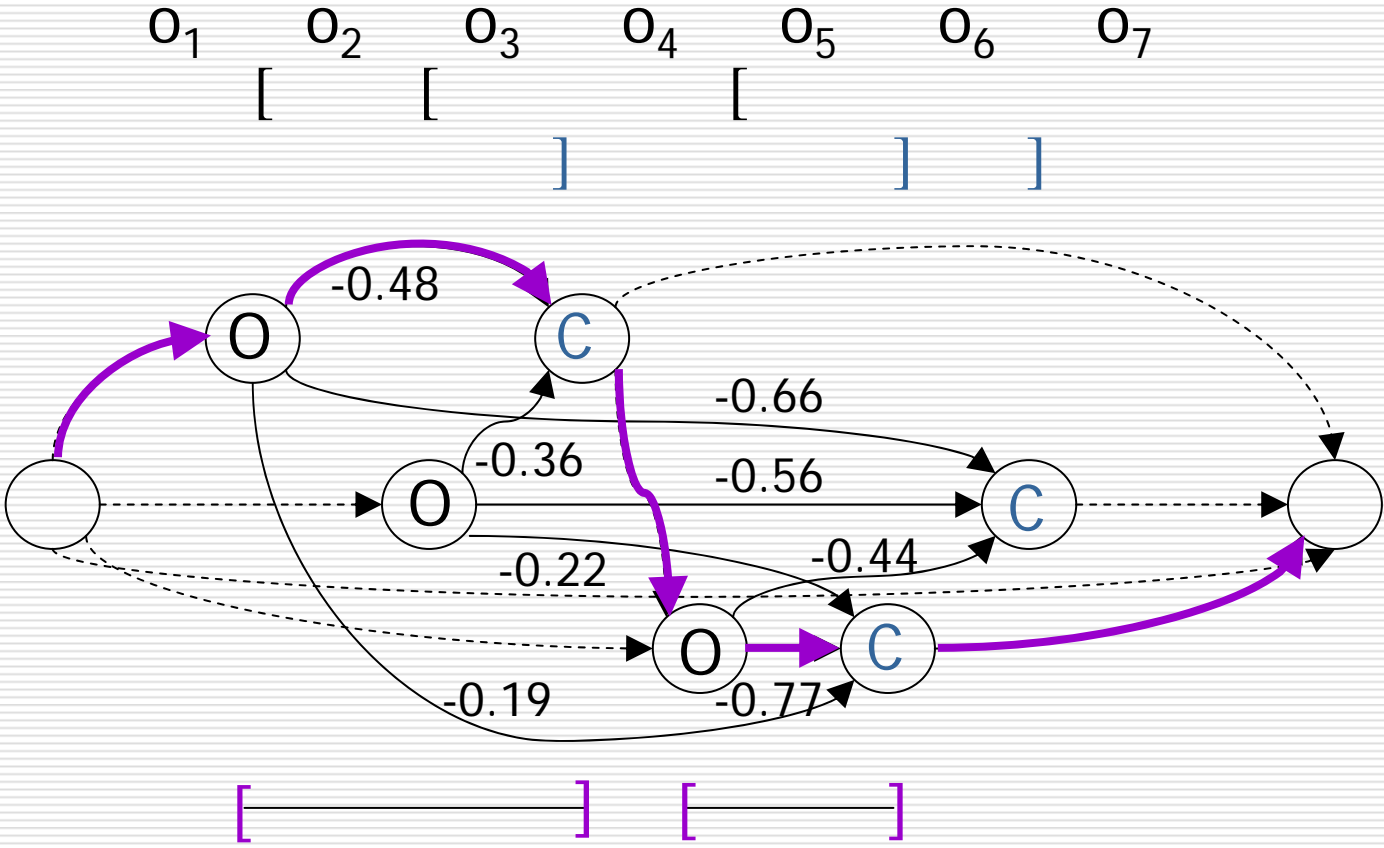
$$(\neg v_i \vee \neg v_j)$$

- Find the solution  $\tau$  that *minimizes the cost*

$$C(\tau) = \sum_{i=1..n} \tau(v_i) c(v_i)$$

- This is a good objective function – it maximizes the expected number of correct chunks.
- CSP in general is hard
- Structure of the constraints yields a problem that can be solved by a shortest path algorithm

# Inference for Shallow Parsing



# Experimental Results

---

## □ NP Prediction

	Recall	Precision	F1
HMM (POS)	89.08	86.62	87.83
HMM with Classifiers (POS)	91.49	91.53	91.51
HMM with Classifiers	93.85	93.46	93.65
PMM	94.28	94.02	94.15
CSCL	94.12	93.45	93.78

# Other Approaches to Shallow Parsing

---

- MEMM
- CRF
- Global Perceptron (Collins')
- Others (e.g., global SVM)

# Maximum Entropy Markov Model (MEMM)

---

- Similar to PMM

$$\begin{aligned} & \arg \max_{s_k, s_{k-1}, \dots, s_1} P(s_k, s_{k-1}, \dots, s_1 | O) \quad \text{for making things dirty} \\ & = \arg \max_{s_k, s_{k-1}, \dots, s_1} \left[ \prod_{t=2}^k P(s_t | s_{t-1}, O) \right] \cdot P(s_1 | O) \end{aligned}$$

- Each term is learned with maximum entropy model

# Conditional Random Field (CRF)

---

- Similar to PMM, but based on Markov random field theory (undirected graph)

$$\arg \max_{s_k, s_{k-1}, \dots, s_1} P(s_k, s_{k-1}, \dots, s_1 | O)$$

$$= \arg \max_{s_k, s_{k-1}, \dots, s_1} \frac{1}{Z(O)} \exp\left(\sum_i w_i f_i(s_k, s_{k-1}, O)\right)$$

# Collins' Perceptron Training for HMM

- Similar to HMM, but use Perceptron algorithm to learn

$$\arg \max_{s_k, s_{k-1}, \dots, s_1} P(s_k, s_{k-1}, \dots, s_1 | O)$$

$$= \arg \max_{s_k, s_{k-1}, \dots, s_1} P(o_k | s_k) \cdot \left[ \prod_{t=1}^{k-1} P(s_{t+1} | s_t) \cdot P(o_t | s_t) \right] \cdot P(s_1)$$

$$= \arg \max_{s_k, s_{k-1}, \dots, s_1} \log(P(o_k | s_k)) \cdot \left[ \prod_{t=1}^{k-1} P(s_{t+1} | s_t) \cdot P(o_t | s_t) \right] \cdot P(s_1)$$

$$= \arg \max_{s_k, s_{k-1}, \dots, s_1} \log(P(o_k | s_k)) + \left[ \sum_{t=1}^{k-1} \log(P(s_{t+1} | s_t)) + \log(P(o_t | s_t)) \right] + \log(P(s_1))$$

$$= \arg \max_{s_k, s_{k-1}, \dots, s_1} w_i f_i(o_k, s_k) + \left[ \sum_{t=1}^{k-1} u_t g_t(s_{t+1}, s_t) + w_t f_t(o_t, s_t) \right] + u_1 g_1(s_1)$$

# Maximum Entropy Markov Model (MEMM)

---

$$P(s_k | s_{k-1}, O) \\ = \frac{1}{Z(s_{k-1}, O)} \exp\left(\sum_i w_i f_i(s_k, s_{k-1}, O)\right)$$

$$\arg \max_{s_k, s_{k-1}, \dots, s_1} P(s_k, s_{k-1}, \dots, s_1 | O)$$

$$= \arg \max_{s_k, s_{k-1}, \dots, s_1} \left[ \prod_{t=2}^k P(s_t | s_{t-1}, O) \right] \cdot P(s_1 | O) \\ = \arg \max_{s_k, s_{k-1}, \dots, s_1} \left[ \prod_{t=2}^k \frac{1}{Z(s_{t-1}, O)} \exp\left(\sum_i w_i f_i(s_t, s_{t-1}, O)\right) \right] \\ \cdot \frac{1}{Z(O)} \exp\left(\sum_i w_i f_i(s_1, O)\right)$$

# Maximum Entropy Markov Model (MEMM)

---

$$\arg \max_{s_k, s_{k-1}, \dots, s_1} P(s_k, s_{k-1}, \dots, s_1 | O)$$

$$\begin{aligned} &= \arg \max_{s_k, s_{k-1}, \dots, s_1} \log \left( \left[ \prod_{t=2}^k \frac{1}{Z(s_{t-1}, O)} \exp \left( \sum_i w_i f_i(s_t, s_{t-1}, O) \right) \right] \right. \\ &\quad \left. \cdot \frac{1}{Z(O)} \exp \left( \sum_i w_i f_i(s_1, O) \right) \right) \\ &= \arg \max_{s_k, s_{k-1}, \dots, s_1} \left[ \sum_{t=2}^k \log \left( \frac{1}{Z(s_{t-1}, O)} \right) + \sum_i w_i f_i(s_t, s_{t-1}, O) \right] \\ &\quad + \log \left( \frac{1}{Z(O)} \right) + \sum_i w_i f_i(s_1, O) \end{aligned}$$

# Conditional Random Field (CRF)

---

$$\arg \max_{s_k, s_{k-1}, \dots, s_1} P(s_k, s_{k-1}, \dots, s_1 | O)$$

$$= \arg \max_{s_k, s_{k-1}, \dots, s_1} \frac{1}{Z(O)} \exp\left(\sum_i w_i f_i(s_k, s_{k-1}, O)\right)$$

$$= \arg \max_{s_k, s_{k-1}, \dots, s_1} \log\left(\frac{1}{Z(O)} \exp\left(\sum_i w_i f_i(s_k, s_{k-1}, O)\right)\right)$$

$$= \arg \max_{s_k, s_{k-1}, \dots, s_1} \log\left(\frac{1}{Z(O)}\right) + \sum_i w_i f_i(s_k, s_{k-1}, O)$$

# Conclusions

---

- All approaches use linear representation
- The differences are
  - Features
  - How to learn weights
  - Training Paradigms:
    - Global Training (HMM, CRF, Global Perceptron)
    - Modular Training (PMM, MEMM, CSCL)
      - These approaches are easier to train, but may requires additional mechanisms to enforce global constraints.