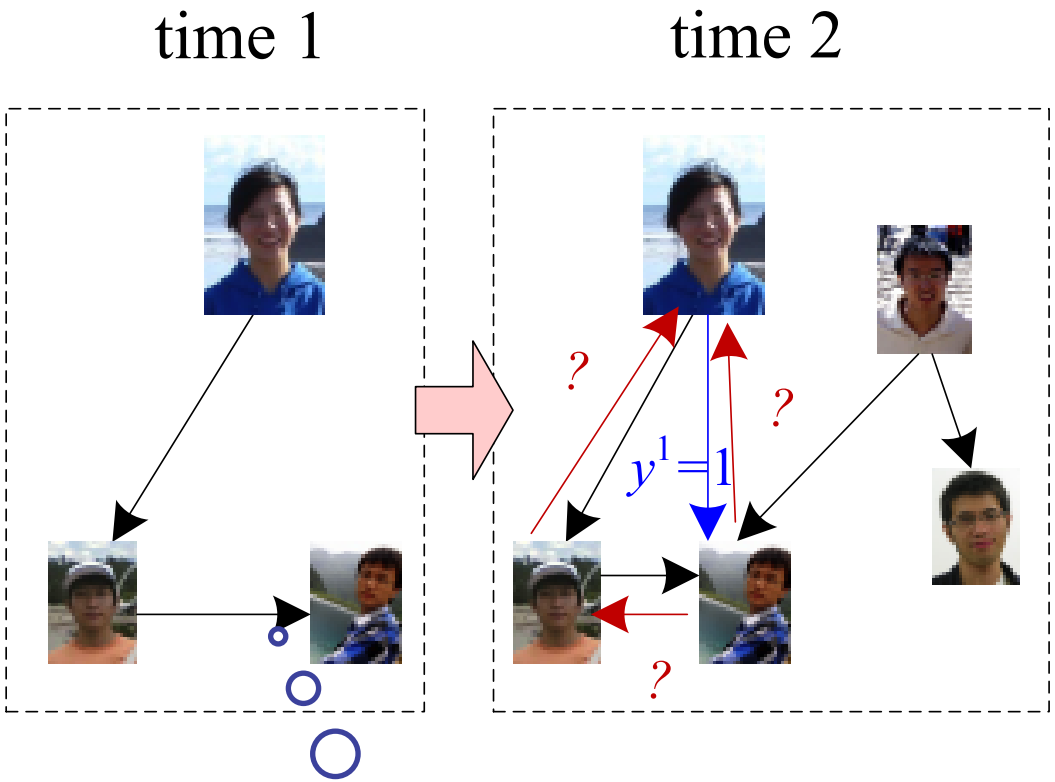
Conditional Random Fields

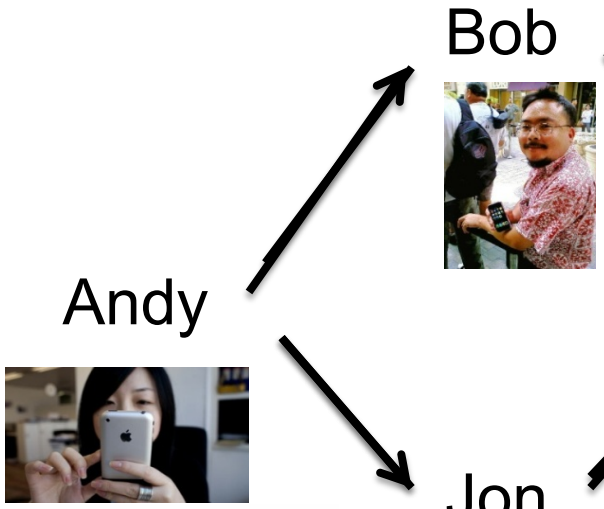
Jie Tang
Lecture for Knowledge Engineering
Department of Computer Science and Technology
Tsinghua University

Follow back Prediction



When you follow a friend in Twitter, how likely he will follow back?

Retweet Predicting



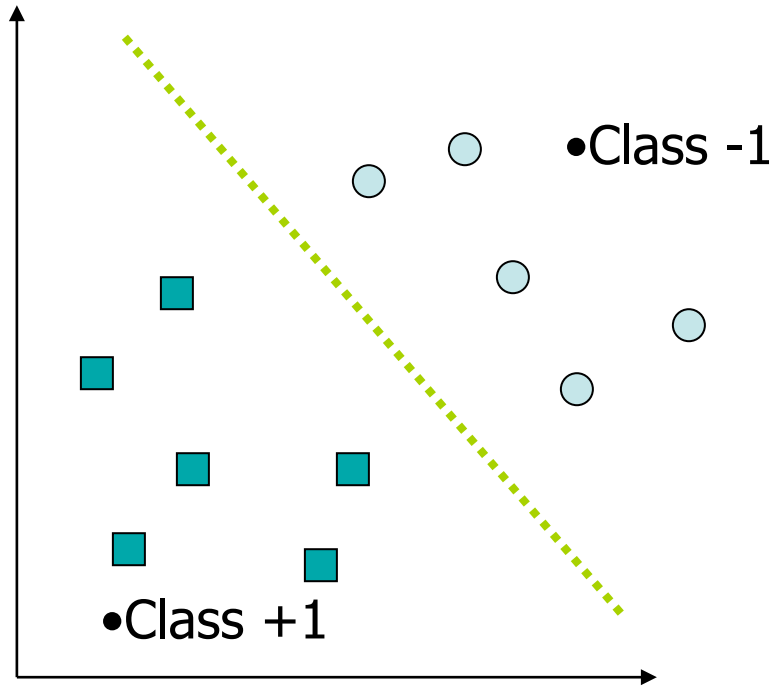


When you post a tweet…





Binary Classifier



Sequence Labeling

- Pos Tagging
  - E.g. [He/PRP] [reckons/VBZ] [the/DT] [current/JJ] [account/NN] [deficit/NN] [will/MD] [narrow/VB] [to/TO] [only/RB] [#/#] [1.8/CD] [billion/CD] [in/IN] [September/NNP] [./.]

- Term Extraction
  - 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.

IE from Web Page

October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open -source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying…

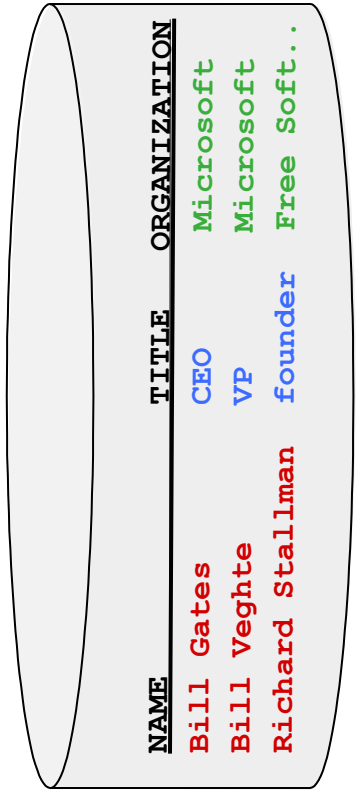
* Microsoft Corporation
CEO
Bill Gates

* Microsoft
Gates

* Microsoft
Bill Veghte

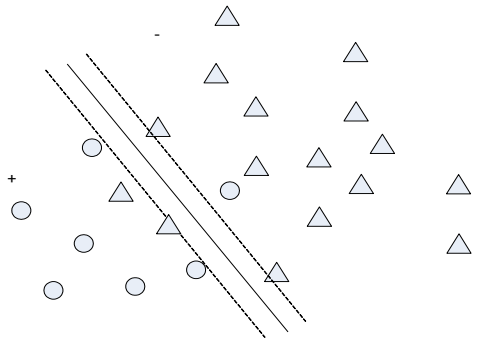
* Microsoft
VP

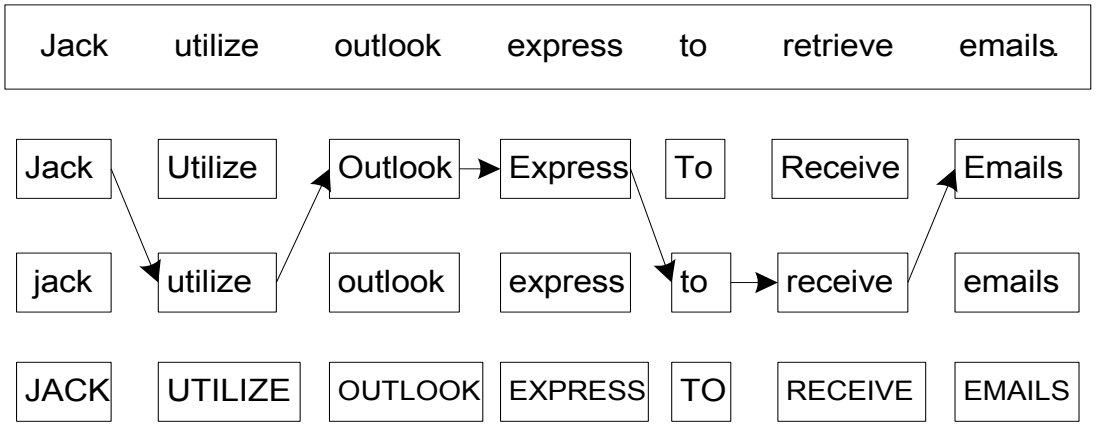
Richard Stallman
founder
Free Software Foundation



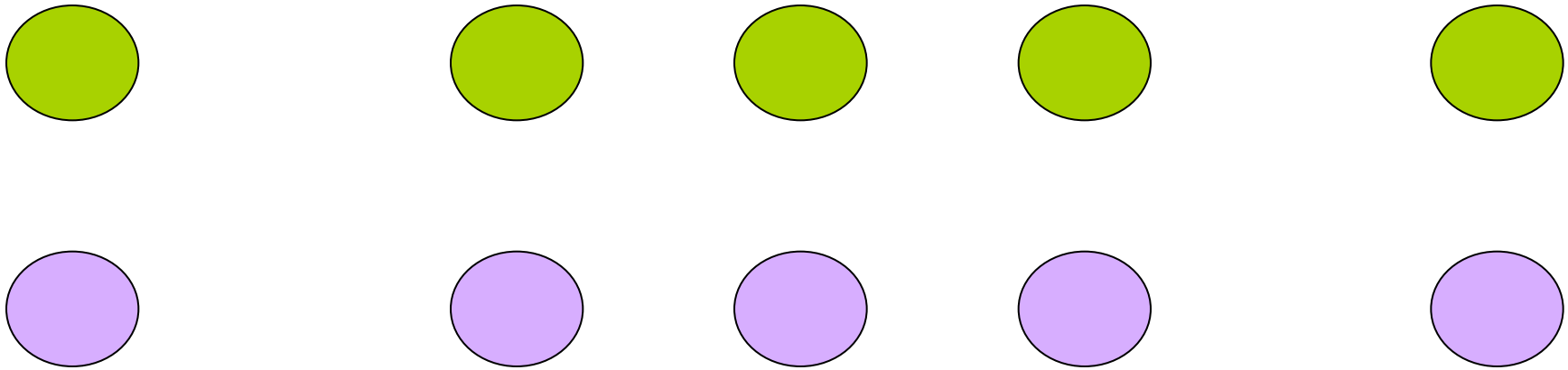
Binary Classifier vs. Sequence Labeling

- Case restoration
  - "jack utilize outlook express to retrieve emails"
  - E.g. SVMs vs. CRFs



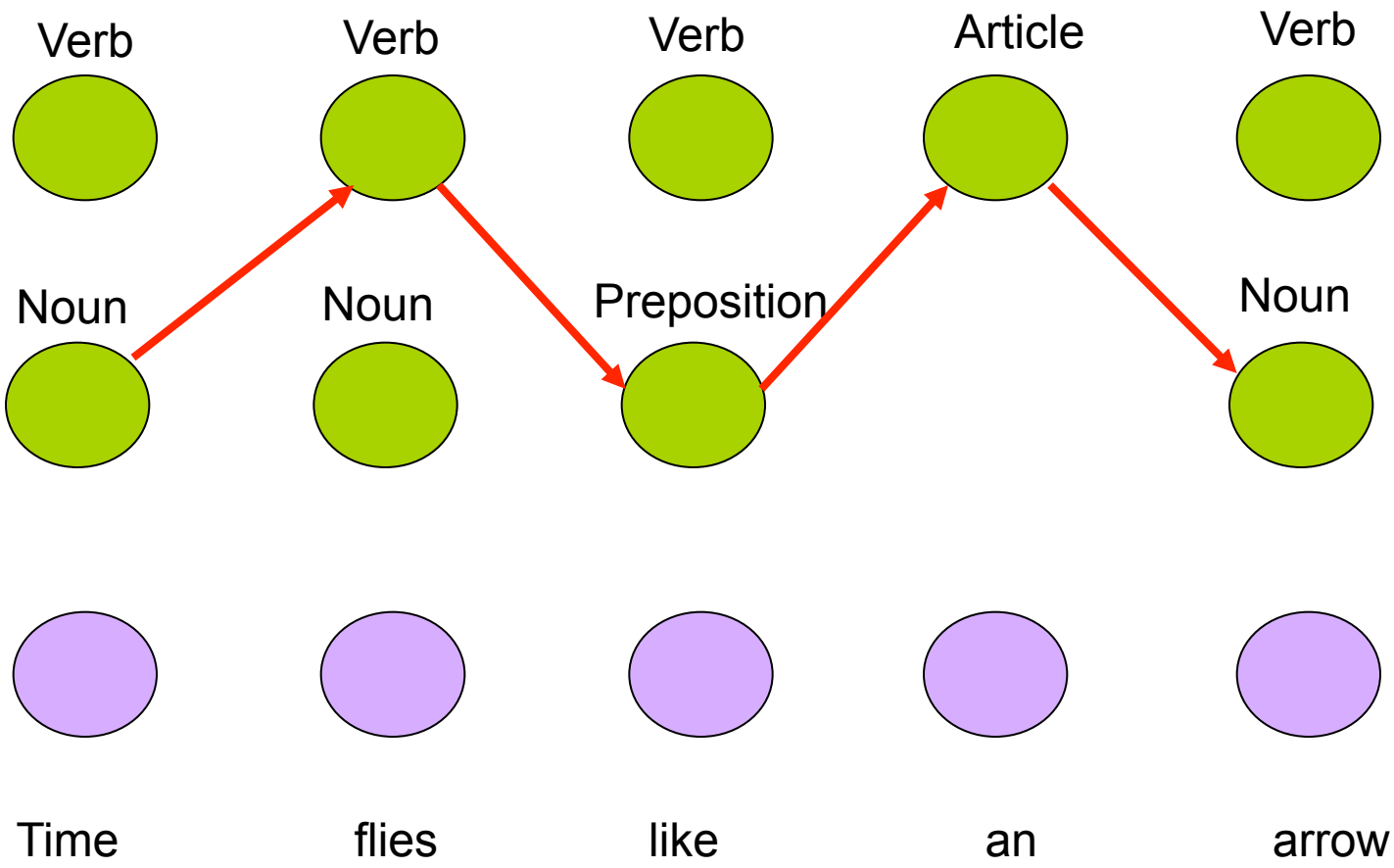


Sequence Labeling Problem

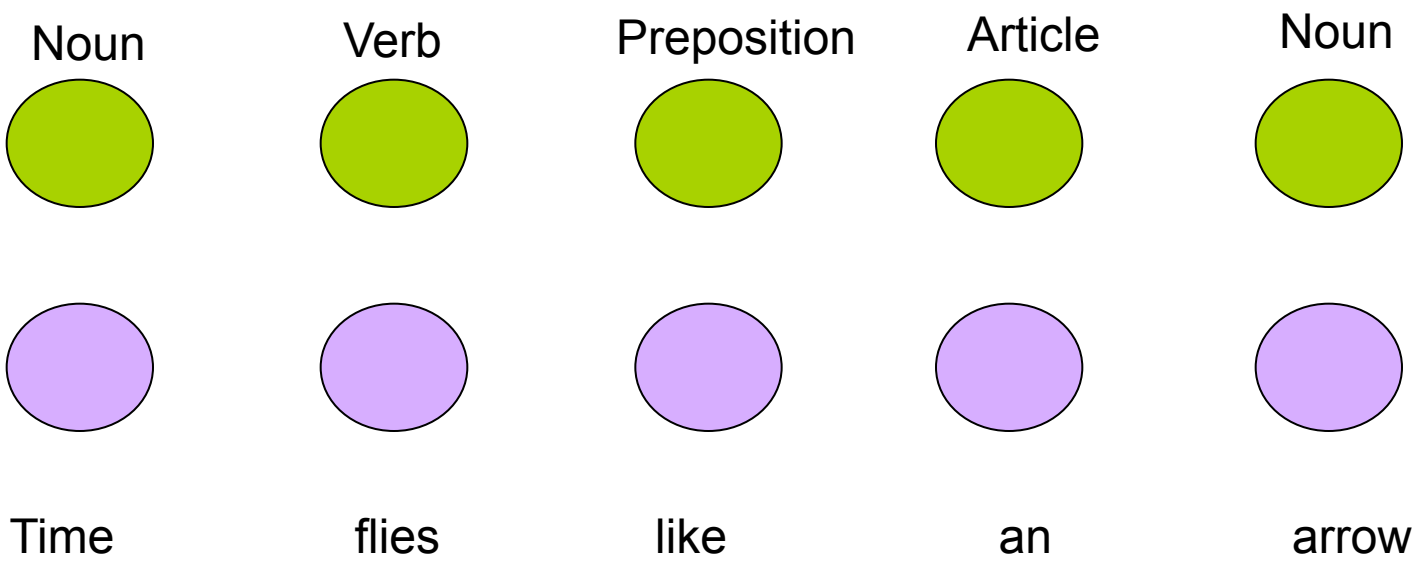


- Green nodes are ***states***
- Purple nodes are ***observations***

Example: POS Tagging Problem



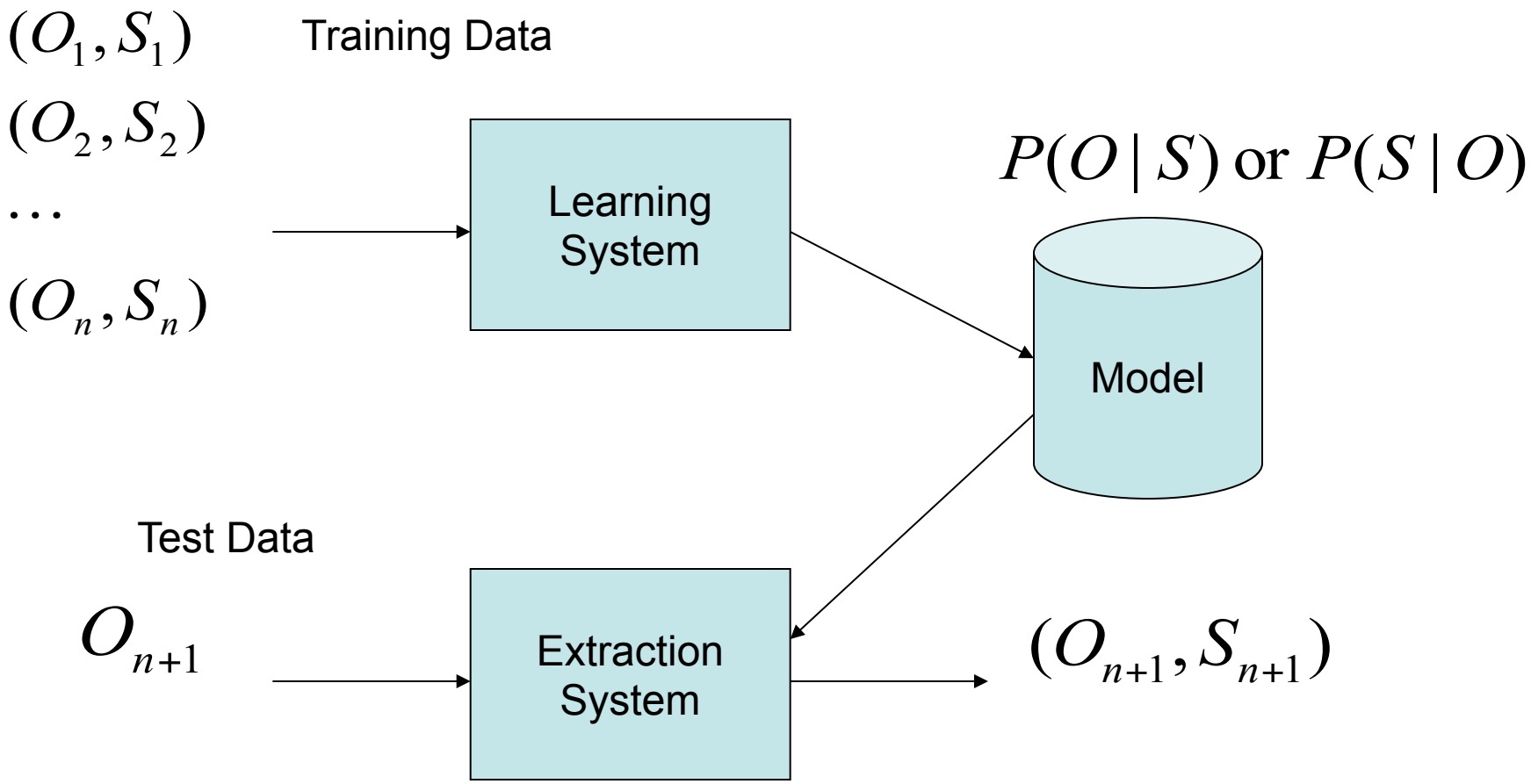
Example: POS Tagging Problem



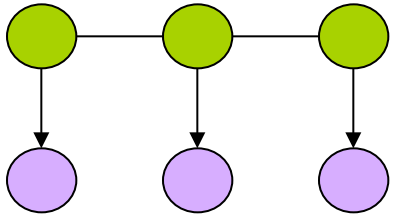
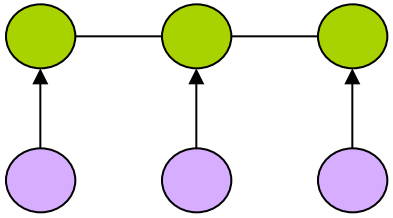
Sequence Labeling Models

- HMM
  - Generative model
  - E.g. Ghahramani (1997), Manning and Schutze (1999)
- MEMM
  - Conditional model
  - E.g. Berger and Pietra (1996), McCallum and Freitag (2000)
- CRFs
  - Conditional model without label bias problem
  - Linear-Chain CRFs
    - E.g. Lafferty and McCallum (2001), Wallach (2004)
  - Non-Linear Chain CRFs
    - Modeling more complex interaction between labels: DCRFs, 2D-CRFs, TCRFs
    - E.g. Sutton and McCallum (2004), Zhu and Nie (2005), Tang et al. (2006)

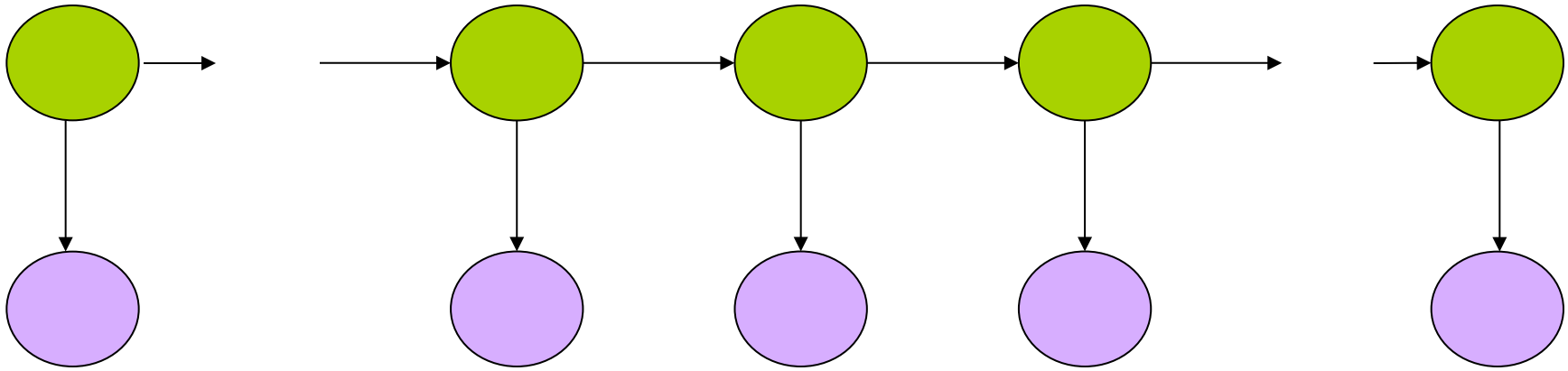
General Framework



Generative vs. Discriminative

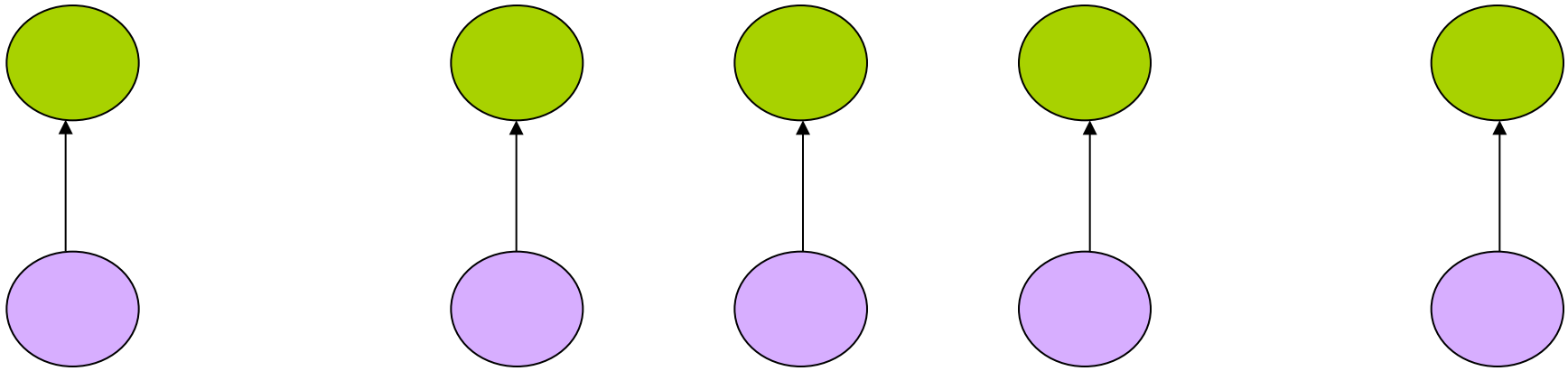
Generative	Discriminative
$P(O \mid S)$	$P(S \mid O)$
Example: HMM	Example: MaxEnt, MEMM, CRF
 States generates observations	 Observations (features) determine states
Learning = finding model generating observation sequence from state sequence	Learning = finding model mapping observation sequence to state sequence
Tagging = finding most likely state sequence having generated given observation sequence	Tagging = finding most likely state sequence mapped from given observation sequence

Assumption 1: Generative Locally Dependent Model



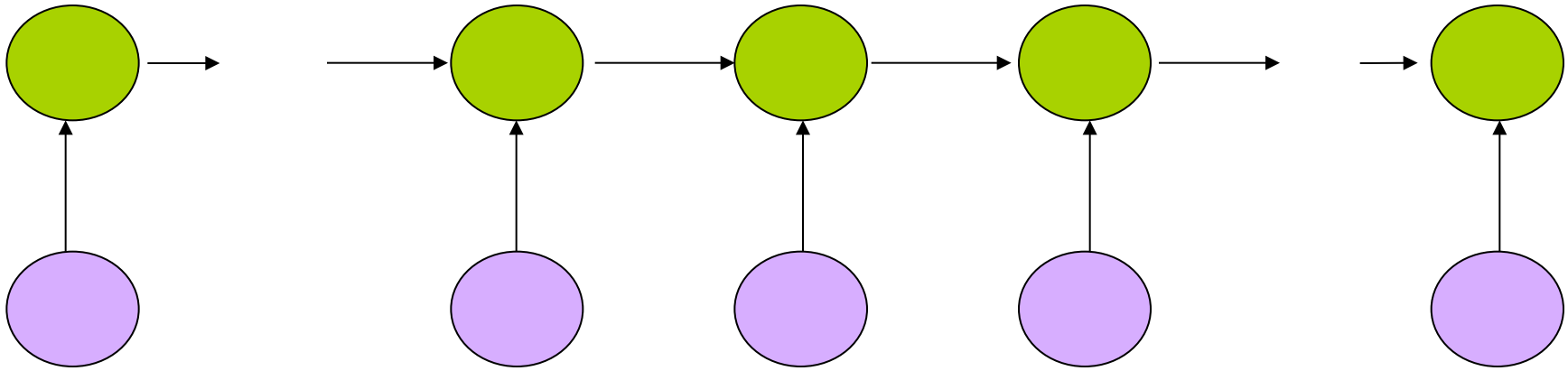
Hidden Markov Model (HMM)

Assumption 2: Discriminative Independent Model



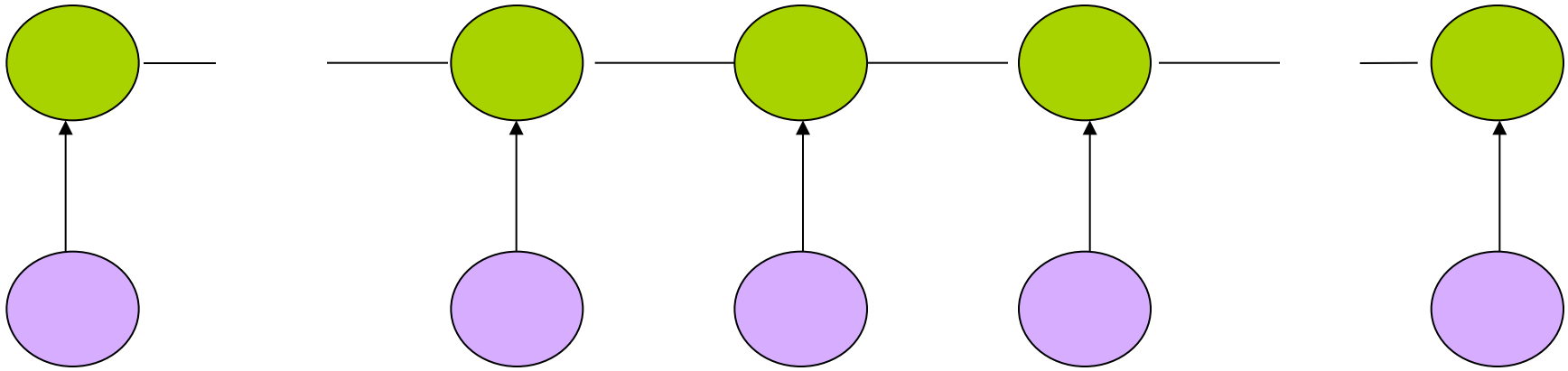
Classifier:
Maximum Entropy Model (ME)
Support Vector Machines (SVM)

Assumption 3: Discriminative Locally Dependent Model



Maximum Entropy Markov Model (MEMM)

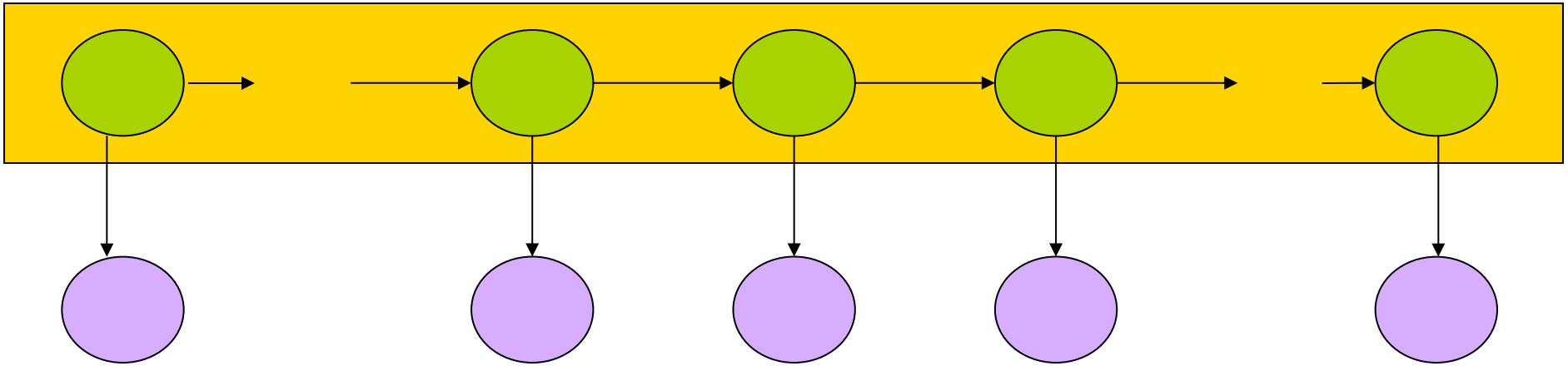
Assumption 4: Discriminative Globally Dependent Model



Conditional Random Field (CRF)

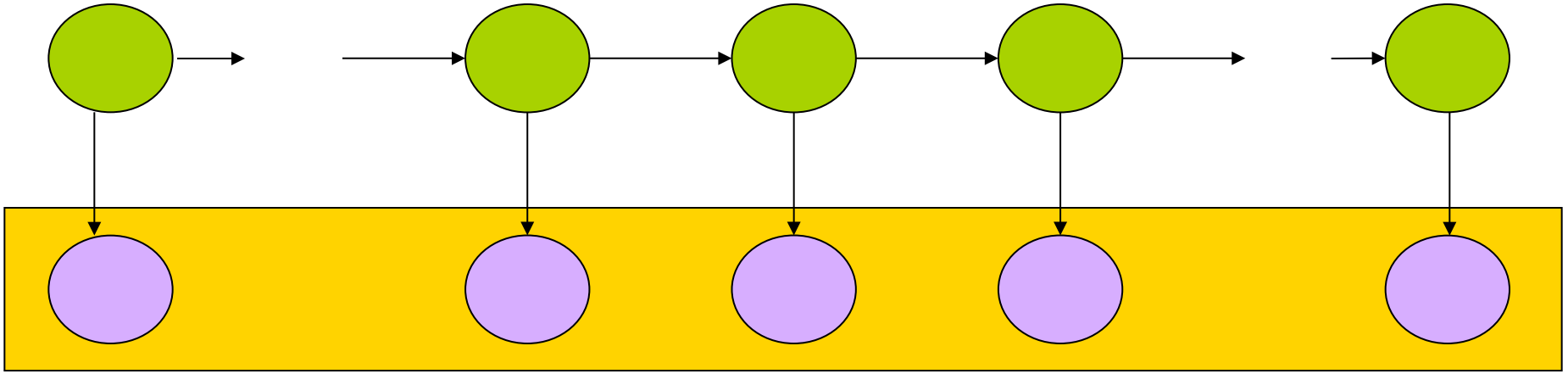
HMM

What is HMM?



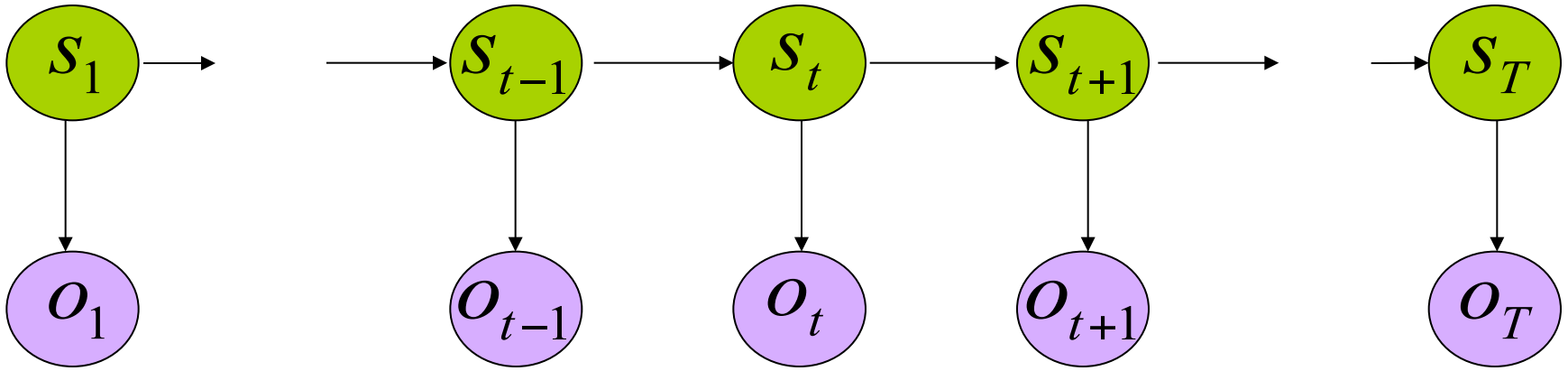
- Green nodes are '***hidden' states***
- State depends only on previous state

What is HMM?



- Purple nodes are ***observations***
- Each state generates an observation

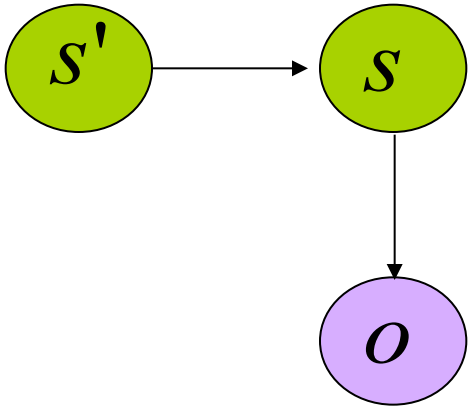
HMM Formalism



- s : {1,2,…,N} are values of hidden states
- o : {1,2,…,M} are values of observations

$$P(O \mid S) = P(s_1)P(o_1 \mid s_1)\prod_{t=2}^{T} P(s_t \mid s_{t-1})P(o_t \mid s_t)$$

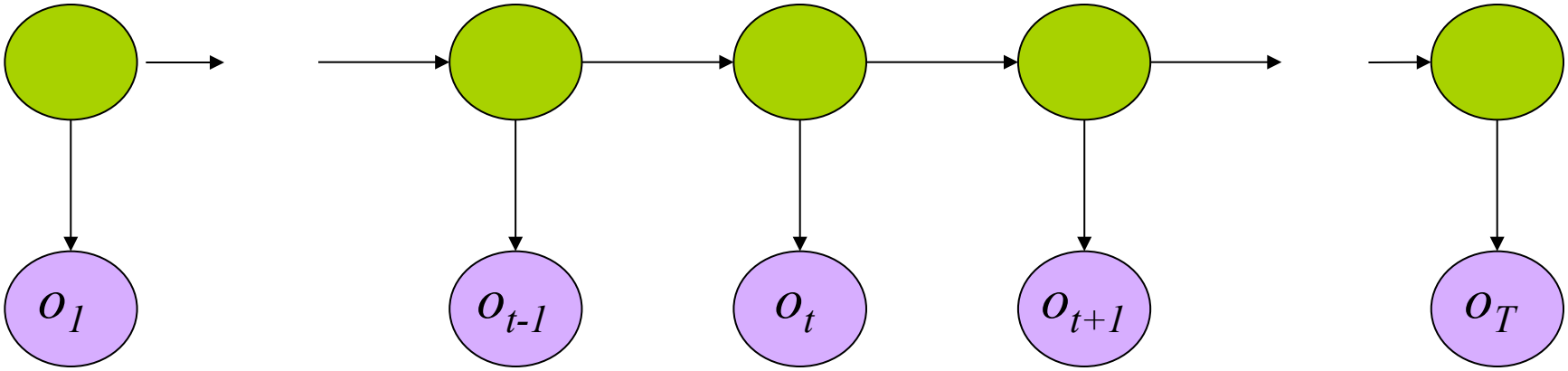
HMM Formalism



$$P(s \mid s')$$

$$P(o \mid s)$$

Tagging



- Viterbi algorithm
  - given observation sequence, compute most likely having generated state sequence

$$\arg\max_S P(S \mid O) = \arg\max_S P(S, O)$$

Summary of HMM

Model

•Baum,1966; Manning, 1999

Applications

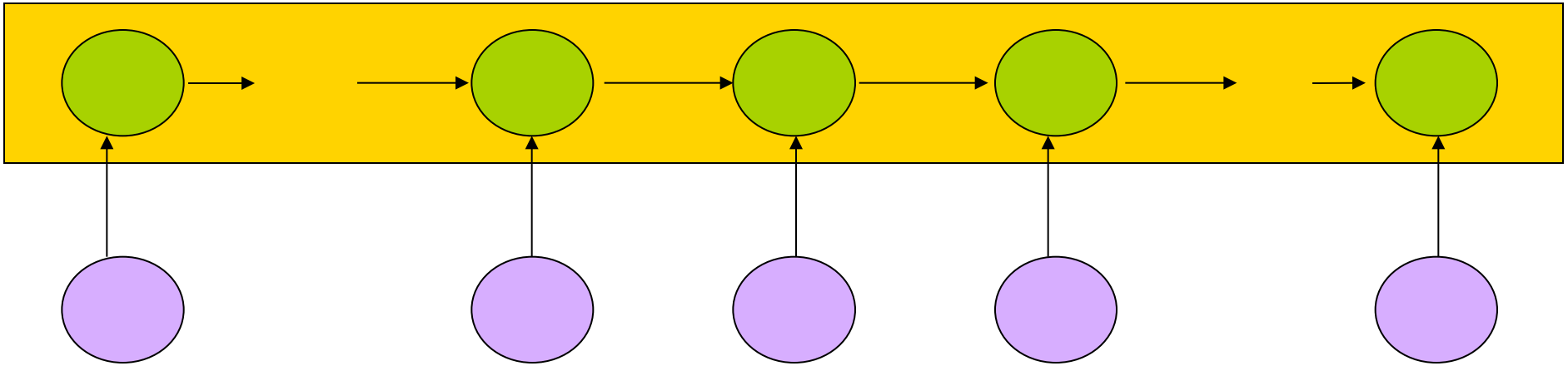
•POS tagging (Kupiec, 1992)
•Shallow parsing (Molina, 2002; Ferran Pla, 2000; Zhou, 2000)
•Speech recognition (Rabiner, 1989; Rabiner 1993)
•Gene sequence analysis (Durbin, 1998)
•…

Limitation

•Joint probability distribution $p(x, s)$.
•Cannot represent overlapping features and long range dependences long range dependences between observed elements.

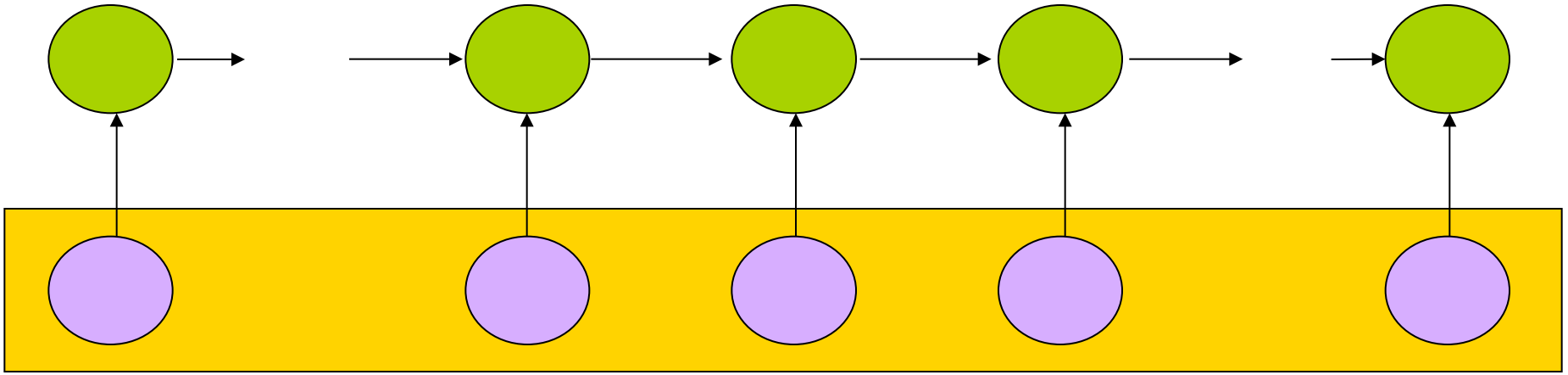
MEMM

What is MEMM?



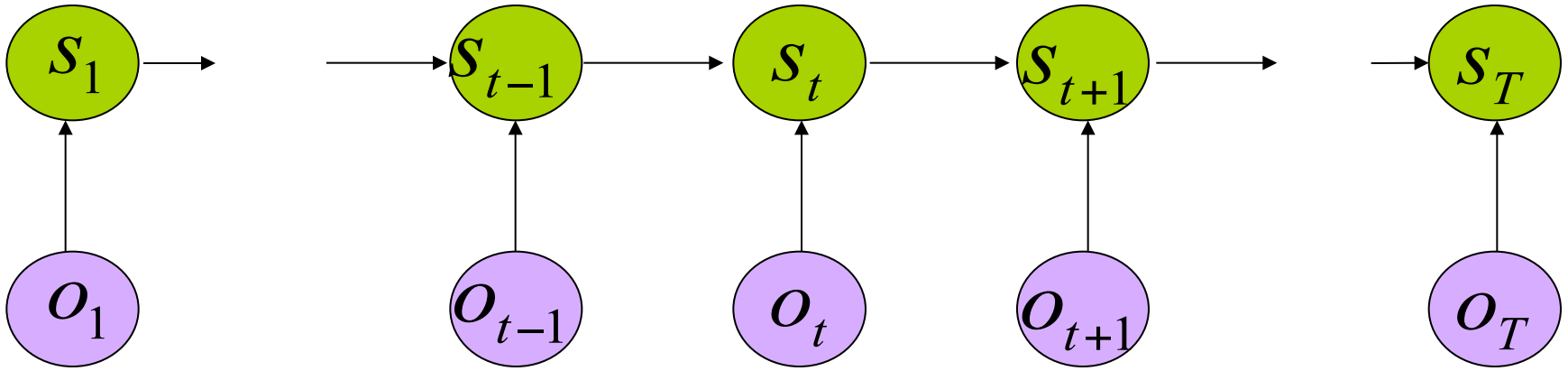
- Green nodes are ***states***
- State depends only on previous state

What is MEMM?



- Purple nodes are ***observations***
- Observations (features of observations) determine states

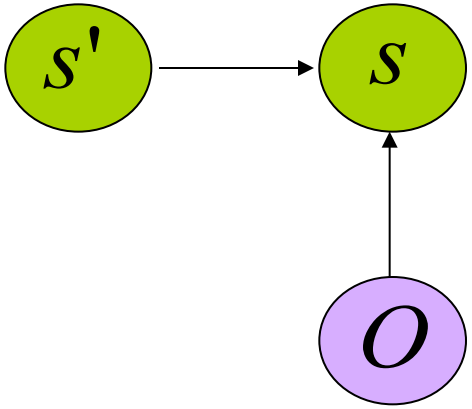
MEMM Formalism



- s : {1,2,…,N } are values for states
- o: {1,2,…,M } are values for observations

$$P(S \mid O) = P(s_1 \mid O) \prod_{t=2}^{T} P(s_t \mid s_{t-1}, O)$$

MEMM Formalism



$$s \leftarrow y, s' \leftarrow y', O \leftarrow x$$

$$P(s \mid s', O) = P(y \mid y', x)$$

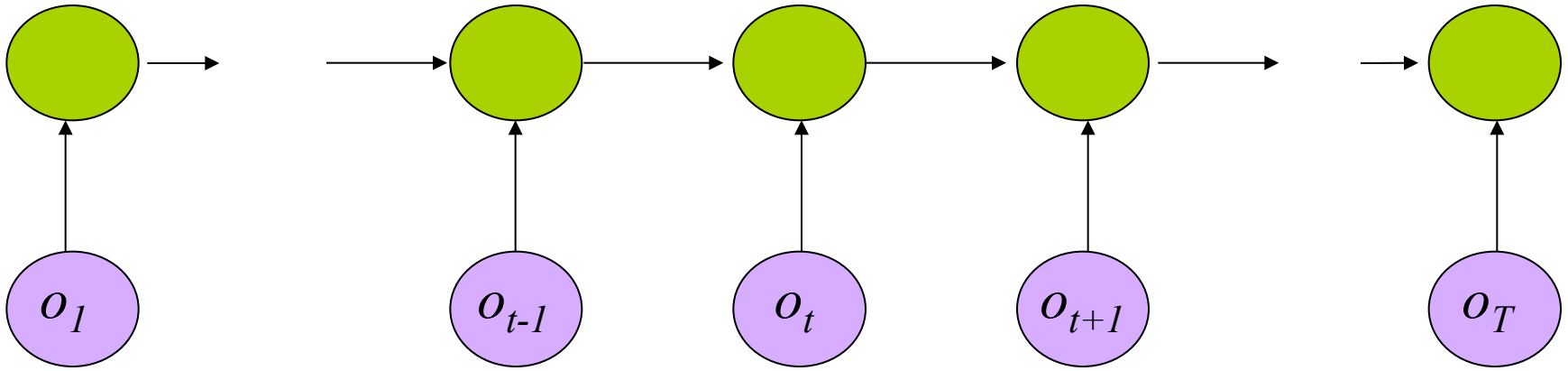
$$P(y \mid y', x) = \frac{\exp(\sum_k \lambda_k f_k(x, y', y))}{Z(y', x)}$$

$$Z(y', x) = \sum_y \exp(\sum_k \lambda_k f_k(x, y', y))$$

Inference in MEMM

- Tagging: given observation sequence, find most likely corresponding state sequence
- Learning: given observation sequence and corresponding state sequence, find model that best explains the matching

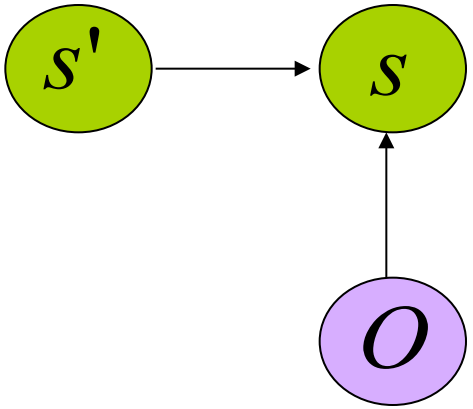
Tagging



- Viterbi algorithm

$$\arg\max_S P(S \mid O) = \arg\max_S P(s_1 \mid O) \prod_{t=1}^{T} P(s_t \mid s_{t-1}, O)$$

Learning



$$(x_1, y_1', y_1), (x_2, y'_2, y_2), \cdots, (x_n, y'_n, y_n)$$

$$P(y \mid y', x) = \frac{\exp(\sum_k \lambda_k f_k(x, y', y))}{Z(x, y')}$$

$$Z(x, y') = \sum_y \exp(\sum_k \lambda_k f_k(x, y', y))$$

$$\arg\max \sum_{i=1}^{n} \log P(y_i \mid y'_i, x_i)$$

Learning Algorithm: IIS

Algorithm 1 : Improved Iterative Scaling

Input: Feature functions $f_1, f_2, \ldots f_n$; empirical distribution $\tilde{p}(x, y)$
Output: Optimal parameter values $\lambda^\star{}_i$; optimal model $p_{\lambda^\star}$

1. Start with $\lambda_i = 0$ for all $i \in \{1, 2, \ldots, n\}$
2. Do for each $i \in \{1, 2, \ldots, n\}$:
   a. Let $\Delta\lambda_i$ be the solution to

$$\sum_{x,y} \tilde{p}(x)p(y|x)f_i(x,y)\exp\Big(\Delta\lambda_i f^{\#}(x,y)\Big) = \tilde{p}(f_i) \tag{16}$$

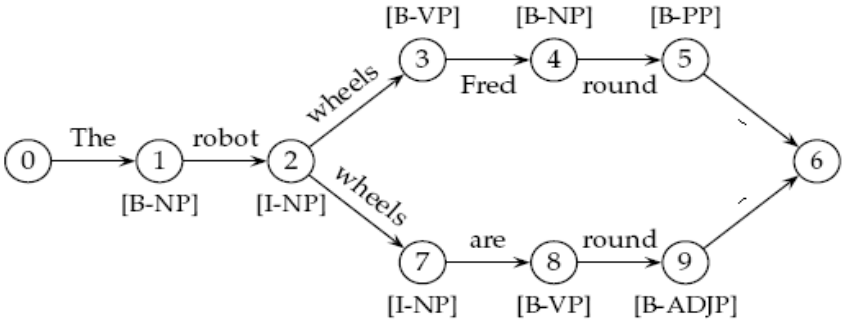
$$\text{where} \quad f^{\#}(x,y) \equiv \sum_{i=1}^{n} f_i(x,y) \tag{17}$$

   b. Update the value of λ_i according to: $\lambda_i \leftarrow \lambda_i + \Delta\lambda_i$
3. Go to step 2 if not all the λ_i have converged

Summary of MEMM

- Discriminative model
- Conditional assumption
- Accuracy is higher than MaxEnt, lower than CRF
- Problem: local model → label bias problem
- MEMM contains MaxEnt as special case

Label Bias Problem



The finite-state acceptor is designed to shallow parse the sentences (chunk/phrase parsing)
1) the robot wheels Fred round
2) the robot wheels are round

Decoding it by:
0123456
0127896

$$p(s \mid x) = p(s_1 \mid x_1) \prod_{i=2}^{n} p(s_i \mid s_{i-1}, x_i)$$

Assuming the probabilities of each of the transitions out of state 2 are approximately equal, the label bias problem means that the probability of each of these chunk sequences given an observation sequence *x* will also be roughly equal irrespective of the observation sequence *x*.

On the other hand, had one of the transitions out of state 2 occurred more frequently in the training data, the probability of that transition would always be greater. This situation would result in the sequence of chunk tags associated with that path being preferred irrespective of the observation sentence.

Summary of MEMM

Model

•Berger, 1996; Ratnaparkhi 1997, 1998

Applications

•Segmentation (McCallum, 2000)

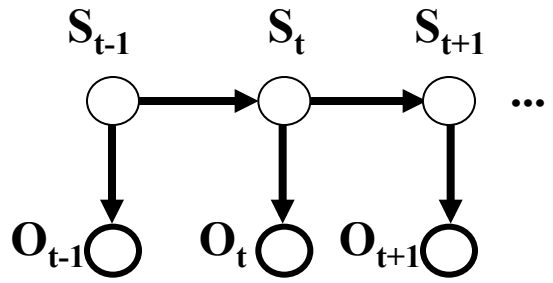
•…

Limitation

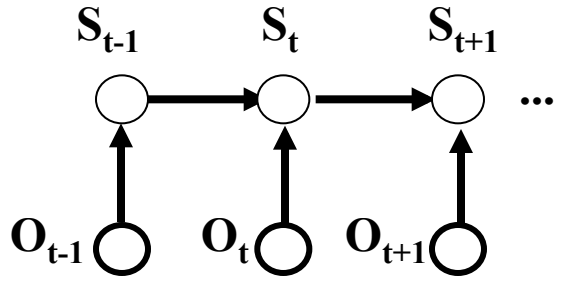
•Label bias problem (HMM do not suffer from the label bias problem)

Conditional Markov Models (CMMs) aka MEMMs aka Maxent Taggers *vs* HMMS

$$\Pr(s,o) = \prod_i \Pr(s_i \mid s_{i-1}) \Pr(o_i \mid s_{i-1})$$



$$\Pr(s \mid o) = \prod_i \Pr(s_i \mid s_{i-1}, o_{i-1})$$



CRFs

MEMM to CRFs

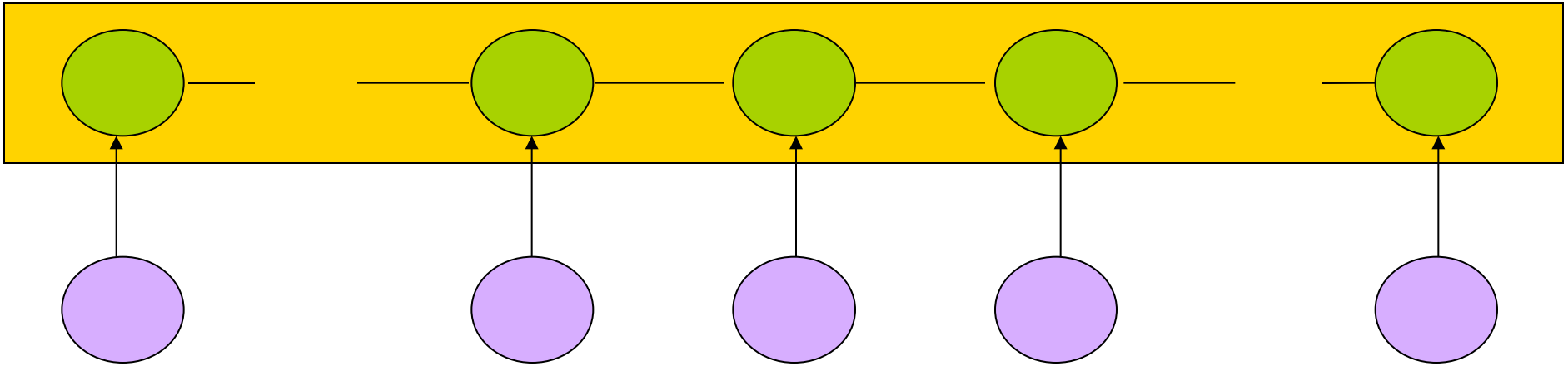
$$\Pr(y_1 \ldots y_n \mid x_1 \ldots x_n) = \prod_j \Pr(y_j \mid y_{j-1}, x_j) = \prod_j \frac{\exp(\sum_i \lambda_i f_i(x_j, y_j, y_{j-1}))}{Z(x_j)}$$

$$= \frac{\exp(\sum_i \lambda_i F_i(\vec{x}, \vec{y}))}{\prod_j Z(x_j)}, \text{ where } F_i(\vec{x}, \vec{y}) = \sum_j f_i(x_j, y_j, y_{j-1})$$

New model

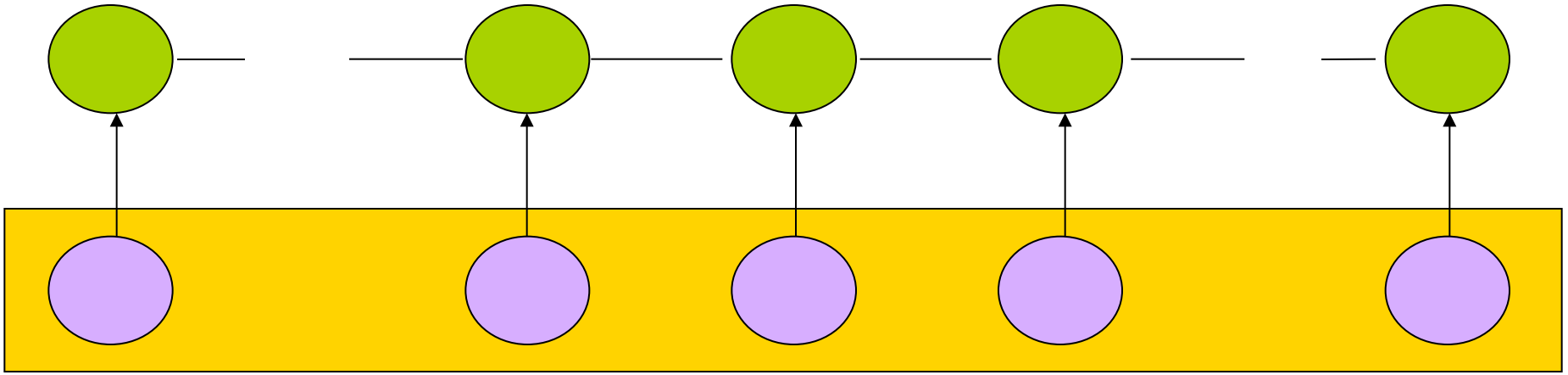
$$\frac{\exp(\sum_i \lambda_i F_i(\vec{x}, \vec{y}))}{Z(\vec{x})}$$

What is CRF?



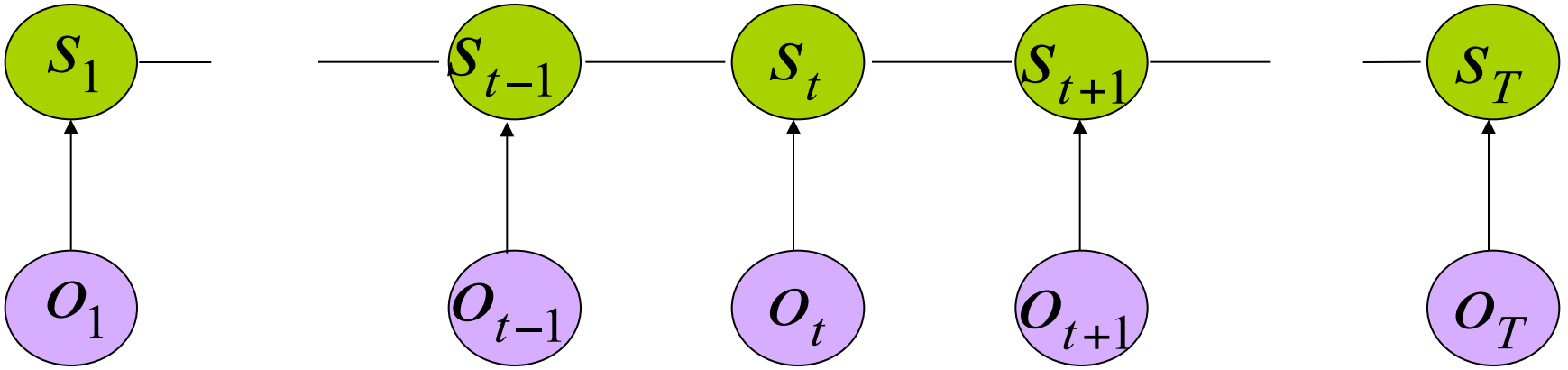
- Green nodes are ***states***
- State depends on neighboring states

What is CRF?



- Purple nodes are ***observations***
- Observations (features of observations) determine states

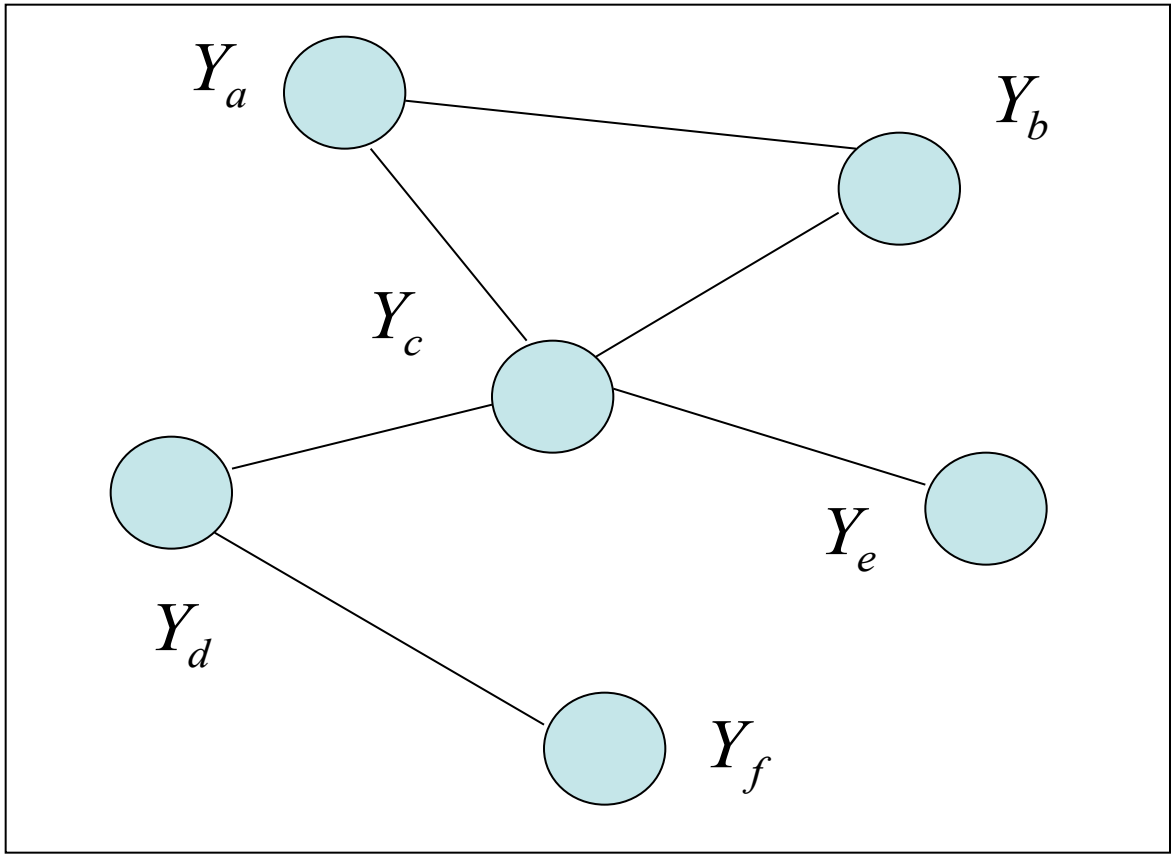
CRF Formalism



- *s*: {1,2,…,N} are values of states
- *o*: {1,2,…,M } are values of observations

$$P(S \mid O)$$

Random Field



Given an undirected graph $G=(V, E)$ such that $Y=\{Y_v | v \in V\}$, if

the probability of Y_v given X and those random variables corresponding to nodes neighboring v in G. Then (X, Y) is a conditional random field.

undirected graphical model globally conditioned on X

$$p(Y_v \mid X, Y_u, u \neq v, \{u,v\} \in V) \Leftrightarrow p(Y_v \mid X, Y_u, (u,v) \in E)$$

Definition

CRF is a Markov Random Fields.

By the Hammersley-Clifford theorem, the probability of a label can be expressed as a Gibbs distribution, so that

$$p(y \mid x, \lambda, \mu) = \frac{1}{Z} \exp\left(\sum_j \lambda_j F_j(y, x)\right)$$

$$F_j(y, x) = \sum_{i=1}^{n} f_j(y_{|c}, x, i)$$

clique

What is clique?

By only taking consideration of the one node and two nodes cliques, we have

$$p(y \mid x, \lambda, \mu) = \frac{1}{Z} \exp\left(\sum_j \lambda_j t_j(y_{|e}, x, i) + \sum_k \mu_k s_k(y_{|s}, x, i)\right)$$

Definition (cont.)

Moreover, let us consider the problem in a first-order chain model, we have

$$p(y \mid x, \lambda, \mu) = \frac{1}{Z} \exp\left(\sum_j \lambda_j t_j(y_{i-1}, y_i, x, i) + \sum_k \mu_k s_k(y_i, x, i)\right)$$

For simplifying description, let $f_j(y, x)$ denote $t_j(y_{i-1}, y_i, x, i)$ and $s_k(y_i, x, i)$

$$p(y \mid x, \lambda, \mu) = \frac{1}{Z} \exp\left(\sum_j \lambda_j F_j(y, x)\right)$$

$$F_j(y, x) = \sum_{i=1}^{n} f_j(y_{|c}, x, i)$$

In Labeling

- In labeling, the task is to find the label sequence that has the largest probability

$$\hat{y} = \arg\max_{y} p_{\lambda}(y \mid x) = \arg\max_{y}(\lambda \cdot F(y, x))$$

$$p(y \mid x, \lambda, \mu) = \frac{1}{Z}\exp(\sum_{j} \lambda_j F_j(y, x))$$

- Then the key is to estimate the parameter lambda
- Let us first review the optimization formalization

Optimization

- Defining a loss function, that should be convex for avoiding local optimization
- Defining constraints
- Finding a optimization method to solve the loss function
- A formal expression for optimization problem

$$\min_{\theta} f(x)$$

$$s.t. \quad g_i(x) \geq 0, 0 \leq i \leq k$$

$$h_j(x) = 0, 0 \leq j \leq l$$

Loss Function

Empirical loss vs. structural loss

$$L = \sum_k |y - f(x, \lambda)|$$

$$\min L$$

$$L = \|\lambda\| + \sum_k |y - f(x, \lambda)|$$

$$\min L$$

Loss function: Log-likelihood

$$p(y \mid x, \lambda, \mu) = \frac{1}{Z} \exp(\sum_j \lambda_j F_j(y, x))$$

$$L(\lambda) = \sum_k \left[-\log Z + \sum_j \lambda_j F_j(y^{(k)}, x^{(k)}) \right]$$

$$L = \sum_k \left[\lambda \cdot F(y^{(k)}, x^{(k)}) - \log Z(x^{(k)}) \right] - \frac{\|\lambda\|^2}{2\sigma^2} + const$$

IIS Algorithm

First-order numerical optimization

Using Iterative Scaling (GIS, IIS)

- Initialize each λ_j(=0 for example)
- Until convergence
  - Solve $\frac{\delta L}{\delta \lambda_j} = 0$ for each parameter λ_j
  - Update each parameter using λ_j<- $\lambda_j + \Delta\lambda_j$

Parameter estimation

Log-likelihood

$$L(\lambda) = \sum_k \left[-\log Z + \sum_j \lambda_j F_j(y^{(k)}, x^{(k)}) \right]$$

Differentiating the log-likelihood with respect to parameter λ_j

$$\frac{\delta L}{\delta \lambda_j} = E_{p(Y,X)}[F_j(Y,X)] - \sum_k E_{p(Y|x^{(k)},\lambda)}[F_j(Y,x^{(k)})]$$

By adding the model penalty, it can be rewritten as

$$\frac{\delta L}{\delta \lambda_j} = E_{p(Y,X)}[F_j(Y,X)] - \sum_k E_{p(Y|x^{(k)},\lambda)}[F_j(Y,x^{(k)})] - \frac{\lambda}{\sigma^2}$$

$$\frac{\delta L_\lambda}{\delta \lambda_j} = \sum_k \left[F_j(y^{(k)}, x^{(k)}) - \frac{(Z(x^{(k)}))'}{Z(x^{(k)})} \right]$$

$$Z(x^{(k)}) = \sum_y \exp \lambda \cdot F(y, x^{(k)})$$

$$\frac{(Z(x^{(k)}))'}{Z(x^{(k)})} = \frac{\sum_y \left(\exp(\lambda \cdot F(y, x^{(k)})) * F_j(y, x^{(k)}) \right)}{\sum_y \exp \lambda \cdot F(y, x^{(k)})}$$

$$= \sum_y \left(\frac{\exp(\lambda \cdot F(y, x^{(k)})}{\sum_y \exp \lambda \cdot F(y, x^{(k)})} * F_j(y, x^{(k)}) \right)$$

$$= \sum_y \left(p(y \mid x^{(k)}) * F_j(y, x^{(k)}) \right)$$

$$= E_{p(Y|x^{(k)})} F_j(Y, x^{(k)})$$

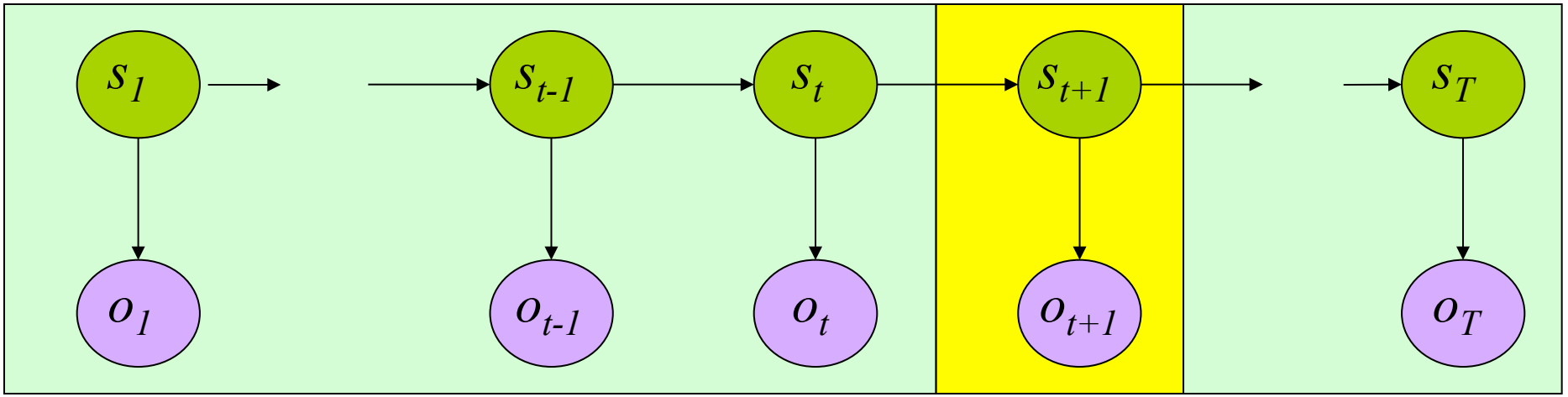
Solve the Optimization

$$L(\lambda) = \sum_k \left[-\log Z + \sum_j \lambda_j F_j(y^{(k)}, x^{(k)}) \right]$$

$$\frac{\delta L}{\delta \lambda_j} = E_{p(Y,X)}[F_j(Y,X)] - \sum_k E_{p(Y|x^{(k)},\lambda)}[F_j(Y, x^{(k)})]$$

- $E_{p(y,x)}F_j(y,x)$ can be calculated easily
- $E_{p(y|x)}F_j(y,x)$ can be calculated by making use of a forward-backward algorithm
- Z can be estimated in the forward-backward algorithm

Forward Backward Algorithm

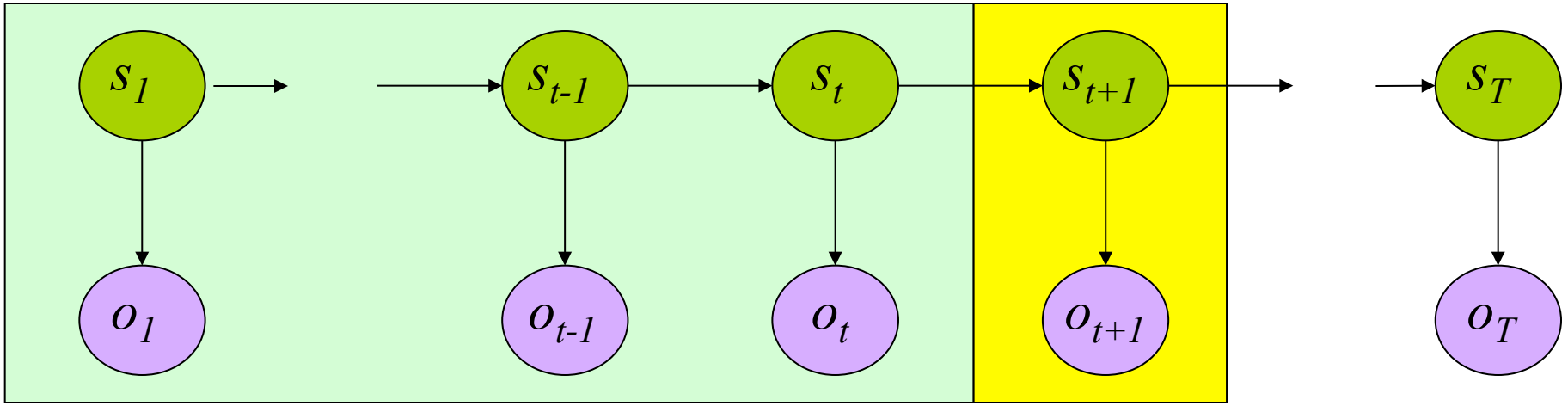


- An efficient algorithm using *dynamic programming*.

$$\alpha_t(i) = P(o_1 \ldots o_t, s_t = i \mid \lambda)$$

$$\beta_t(i) = P(o_t \ldots o_T \mid s_t = i, \lambda)$$

Forward Probability



$$\alpha_t(i) = P(o_1 \ldots o_t, s_t = i \mid \mu)$$

$$\alpha_{t+1}(j) = \sum_{i=1 \ldots N} \alpha_t(i) a_{ij} b_{jo_{t+1}}$$

$$\alpha_1(i) = \pi_i b_{io_1}$$

π are initial state probabilities

Forward Probability

$$\alpha_{t+1}(j)$$

$$= P(o_1 \ldots o_{t+1}, s_{t+1} = j)$$

$$= P(o_1 \ldots o_{t+1} \mid s_{t+1} = j) P(s_{t+1} = j)$$

$$= P(o_1 \ldots o_t \mid s_{t+1} = j) P(o_{t+1} \mid s_{t+1} = j) P(s_{t+1} = j)$$

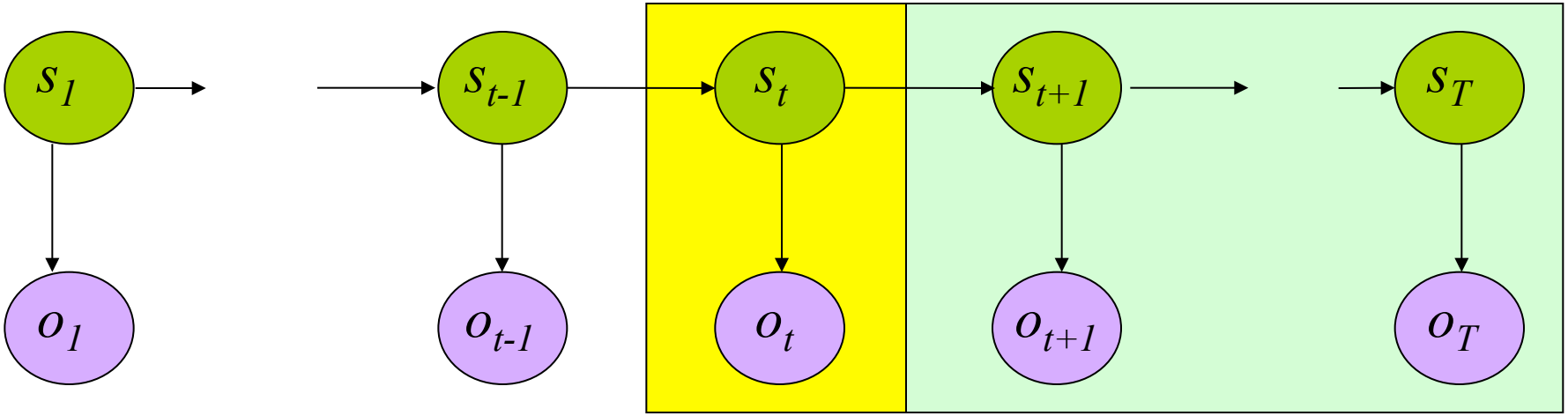
$$= P(o_1 \ldots o_t, s_{t+1} = j) P(o_{t+1} \mid s_{t+1} = j)$$

$$= \sum_{i=1 \ldots N} P(o_1 \ldots o_t, s_t = i, s_{t+1} = j) P(o_{t+1} \mid s_{t+1} = j)$$

$$= \sum_{i=1 \ldots N} P(o_1 \ldots o_t, s_t = i) P(s_{t+1} = j \mid s_t = i) P(o_{t+1} \mid s_{t+1} = j)$$

$$= \sum_{i=1 \ldots N} \alpha_t(i) a_{ij} b_{jo_{t+1}}$$

Backward Probability

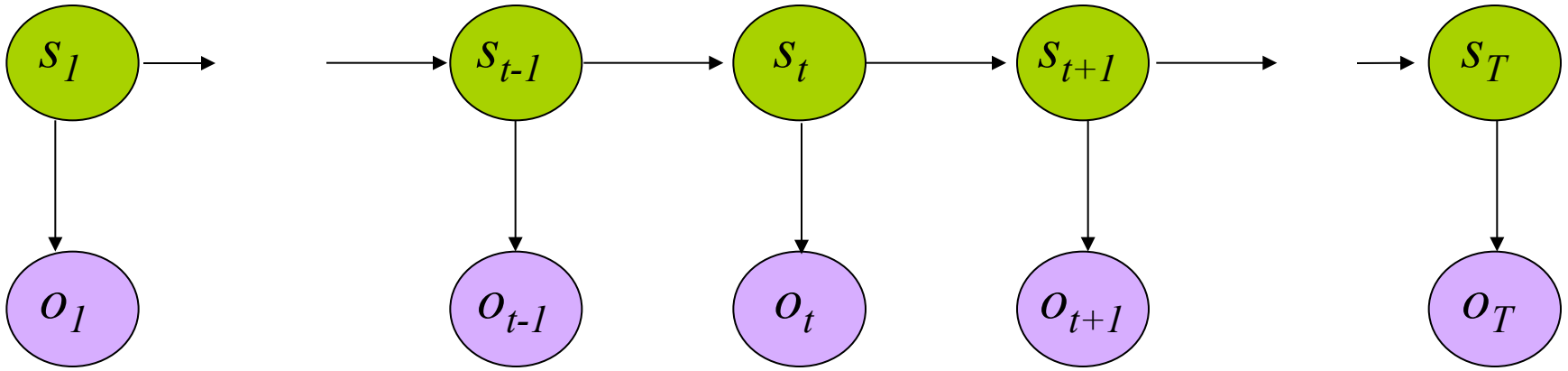


$$\beta_t(i) = P(o_t \ldots o_T \mid s_t = i)$$

$$\beta_{T+1}(i) = 1$$

$$\beta_t(i) = \sum_{j=1\ldots N} a_{ij} b_{io_t} \beta_{t+1}(j)$$

Marginal Probability



$$p_t(i,j) = \frac{\alpha_{t-1}(i) a_{ij} b_{jo_t} \beta_t(j)}{\sum_{k=1\ldots N} \alpha_t(k)\beta_t(k)}$$

$$p_t(i) = \frac{\alpha_t(i)\beta_t(j)}{\sum_{k=1\ldots N} \alpha_t(k)\beta_t(k)}$$

Calculating the Expectation

- First we define the transition matrix of y for position x as

$$M_i[y_{i-1}, y_i] = \exp \lambda \cdot f(y_{i-1}, y_i, x, i)$$

$$E_{p(Y|x^{(k)})}\left[F_j(Y, x^{(k)})\right] = \sum_y p(y \mid x^{(k)}) F_j(y, x)$$

$$= \sum_{i=1}^{n} \sum_{y_{i-1}, y_i} p(y_{i-1}, y_i \mid x^{(k)}) f_j(y_{i-1}, y_i, x^{(k)}) + \sum_{i=1}^{n} \sum_j p(y_i \mid x^{(k)}) f_j(y_i, x^{(k)})$$

$$p(y_{i-1}, y_i \mid x^{(k)}) = \frac{\alpha_{i-1}(M_i * V_i)\beta_i^T}{Z(x)}$$

$$p(y_i \mid x^{(k)}) = \frac{\alpha_i \beta_i^T}{Z(x)}$$

$$Z(x) = \left[\prod_{i=1}^{n+1} M_i(x)\right] = \alpha_n \cdot 1^T$$

$$\alpha_i = \begin{cases} \alpha_i M_i & 0 < i \le n \\ 1 & i = 0 \end{cases}$$

$$\beta_i^T = \begin{cases} M_{i+1}\beta_{i+1}^T & 1 \le i < n \\ 1 & i = n \end{cases}$$

All state features at position *i*

IIS Algorithm

First-order numerical optimization

Using Iterative Scaling (GIS, IIS)

- Initialize each λ_j(=0 for example)
- Until convergence
  - Solve $\frac{\delta L}{\delta \lambda_j} = 0$ for each parameter λ_j
  - Update each parameter using λ_j<- $\lambda_j + \Delta\lambda_j$

Low efficient!!

Second-order numerical optimization

Using newton optimization technique for the parameter estimation

$$\lambda^{(k+1)} = \lambda^{(k)} + (\frac{\partial^2 L}{\partial \lambda^2})^{-1} \frac{\partial L}{\partial \lambda}$$

Drawbacks: parameter value initialization
And compute the second order (i.e. hesse matrix), that is difficult

Solutions:
- Conjugate-gradient (CG) (Shewchuk, 1994)
- Limited-memory quasi-Newton (L-BFGS) (Nocedal and Wright, 1999)
- Voted Perceptron (Colloins 2002)

Summary of CRFs

Model

•Lafferty, 2001

Applications

•Efficient training (Wallach, 2003)
•Training via. Gradient Tree Boosting (Dietterich, 2004)
•Bayesian Conditional Random Fields (Qi, 2005)
•Name entity (McCallum, 2003)
•Shallow parsing (Sha, 2003)
•Table extraction (Pinto, 2003)
•Signature extraction (Kristjansson, 2004)
•Accurate Information Extraction from Research Papers (Peng, 2004)
•Object Recognition (Quattoni, 2004)
•Identify Biomedical Named Entities (Tsai, 2005)
•…

Limitation

•Huge computational cost in parameter estimation

Applications

A Unified Tagging Approach to Text Normalization (ACL'2007)

Conghui Zhu[1], Jie Tang[2], Hang Li[3], Hwee Tou Ng[4], and Tiejun Zhao[1]

[1]Harbin Institute of Technology
[2]Tsinghua University
[3]Microsoft Research Asia
[4]National University of Singapore

Outline

- Motivation
- Related Work
- Problem Description
- A Unified Tagging Approach
- Experimental Results
- Summary

Motivation

- More and more 'informally inputted' text data becomes available to NLP
  - E.g., emails, newsgroups, forums, blogs, etc.
- The informal text is usually very noisy
  - 98.4% of the 5000 randomly selected emails contain noises
- Previously, text normalization is conducted in a more or less ad-hoc manner
  - E.g., heuristic rules or separated classification models

Examples

Noise Text

NER

Case Error

Extra space
Extra punc.
Missing space
Missing period

Extra line break

1. i'm thinking about buying a pocket
2. pc device for my wife this christmas,.
3. the worry that i have is that she won't
4. be able to sync it to her outlook express
5. contacts…

Cannot find any named entities from the noise text

Normalized Text

NER

Product

Date

I'm thinking about buying a Pocket PC device for my wife this Christmas.// The worry that I have is that she won't be able to sync it to her Outlook Express contacts.//

Contain many errors in term extraction

Term Extraction

1. i'm thinking about buying a pocket
2. pc device for my wife this christmas,.
3. the worry that i have is that she won't
4. be able to sync it to her outlook express
5. contacts…

Term Extraction

I'm thinking about buying a Pocket PC device for my wife this Christmas.// The worry that I have is that she won't be able to sync it to her Outlook Express contacts.//

Outline

- Motivation
- Related Work
- Problem Description
- A Unified Tagging Approach
- Experimental Results
- Summary

Related Work – Cleaning Informal Text

- Preprocessing Noisy Texts
  - Clark (2003) and Wong, Liu, and Bennamoun (2006)
- NER from Informal Texts
  - Minkov, Wang, and Cohen (2005)
- Signature Extraction from Informal Text
  - Carvalho and Cohen (2004)
- Email Data Cleaning
  - Tang, Li, Cao, and Tang (2005)

Related Work – Language Processing

- Sentence Boundary Detection
  - E.g., Palmer and Hearst (1997), Mikheev (2000)
- Case Restoration
  - Lita and Ittycheriah (2003), Mikheev (2002)
- Spelling Error Correction
  - Golding and Roth (I996), Brill and Moore (2000), Church and Gale (1991) Mays et al. (1991)
- Word Normalization
  - Sproat, et al. (1999)

Outline

- Motivation
- Related Work
- Problem Description
- A Unified Tagging Approach
- Experimental Results
- Summary

Problem Description

Level	Task	Percentages of Noises
Paragraph	Extra line break deletion	49.53
	Paragraph boundary detection	
Se		
Word	Case restoration	15.04
	Unnecessary token deletion	9.69
	Misspelled word correction	3.41

Text normalization is defined at three levels

An ideal normalization method should consider processing all the tasks together!

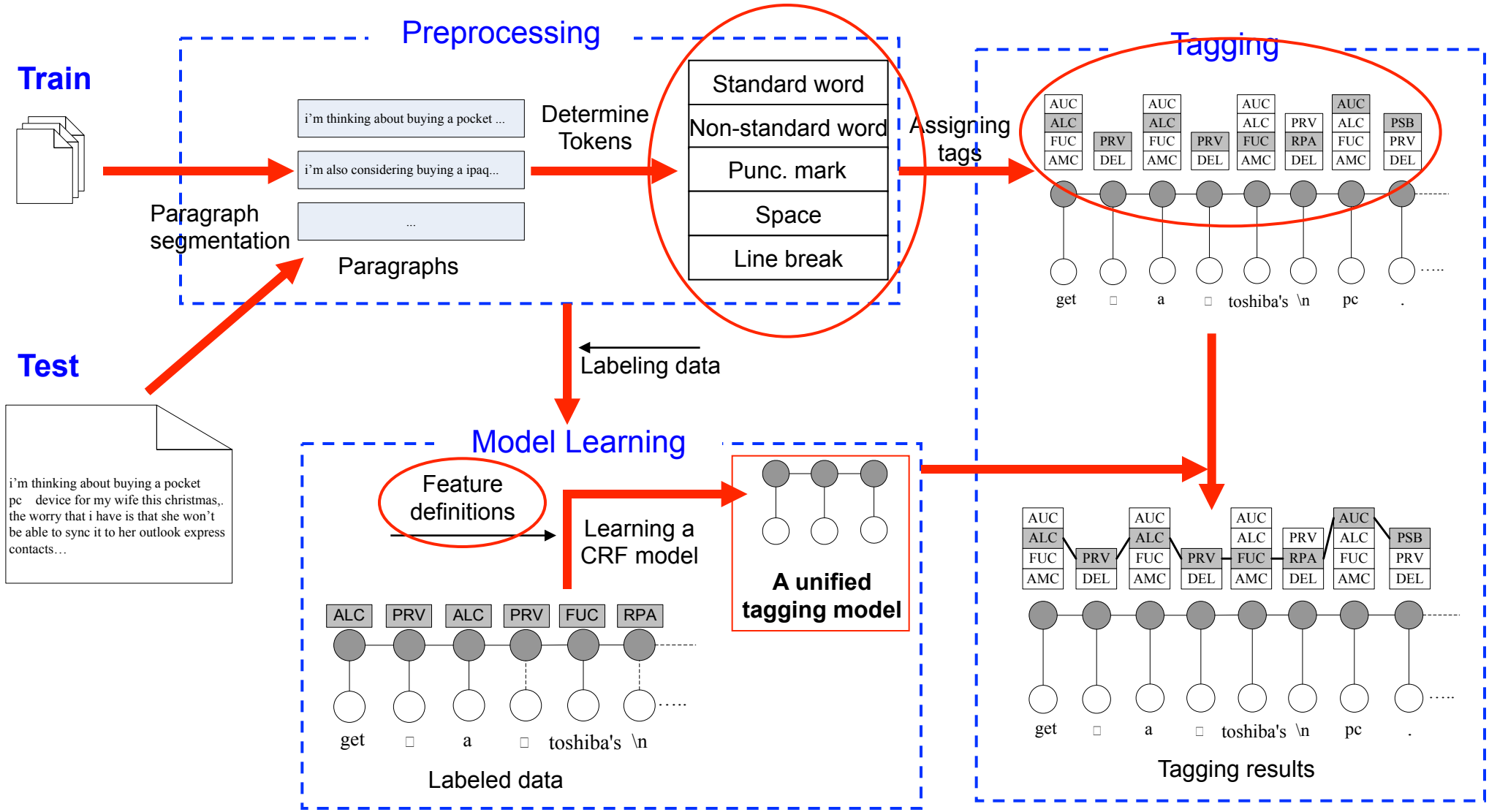
dependencies ween different s of noises

deletion of tokens like '--' and '=='

Outline

- Motivation
- Related Work
- Problem Description
- A Unified Tagging Approach
- Experimental Results
- Summary

Processing Flow

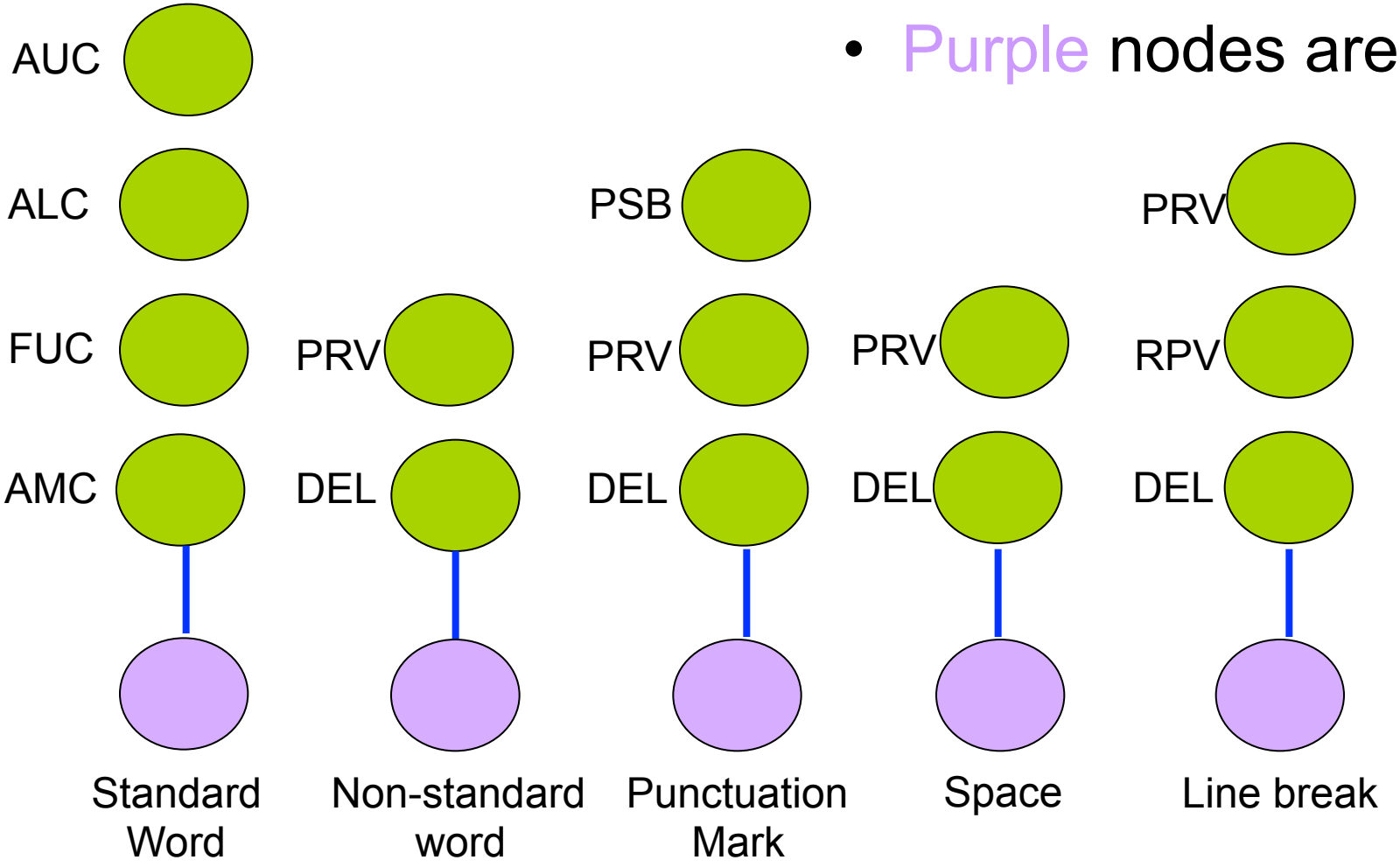


Token Definitions

Standard word	Words in natural language
Non-standard word	Including several general 'special words' e.g. email address, IP address, URL, date, number, money, percentage, unnecessary tokens (e.g. '===' and '###'), etc.
Punctuation marks	Including period, question mark, and exclamation mark
Space	Each space will be identified as a space token
Line break	Every line break is a token

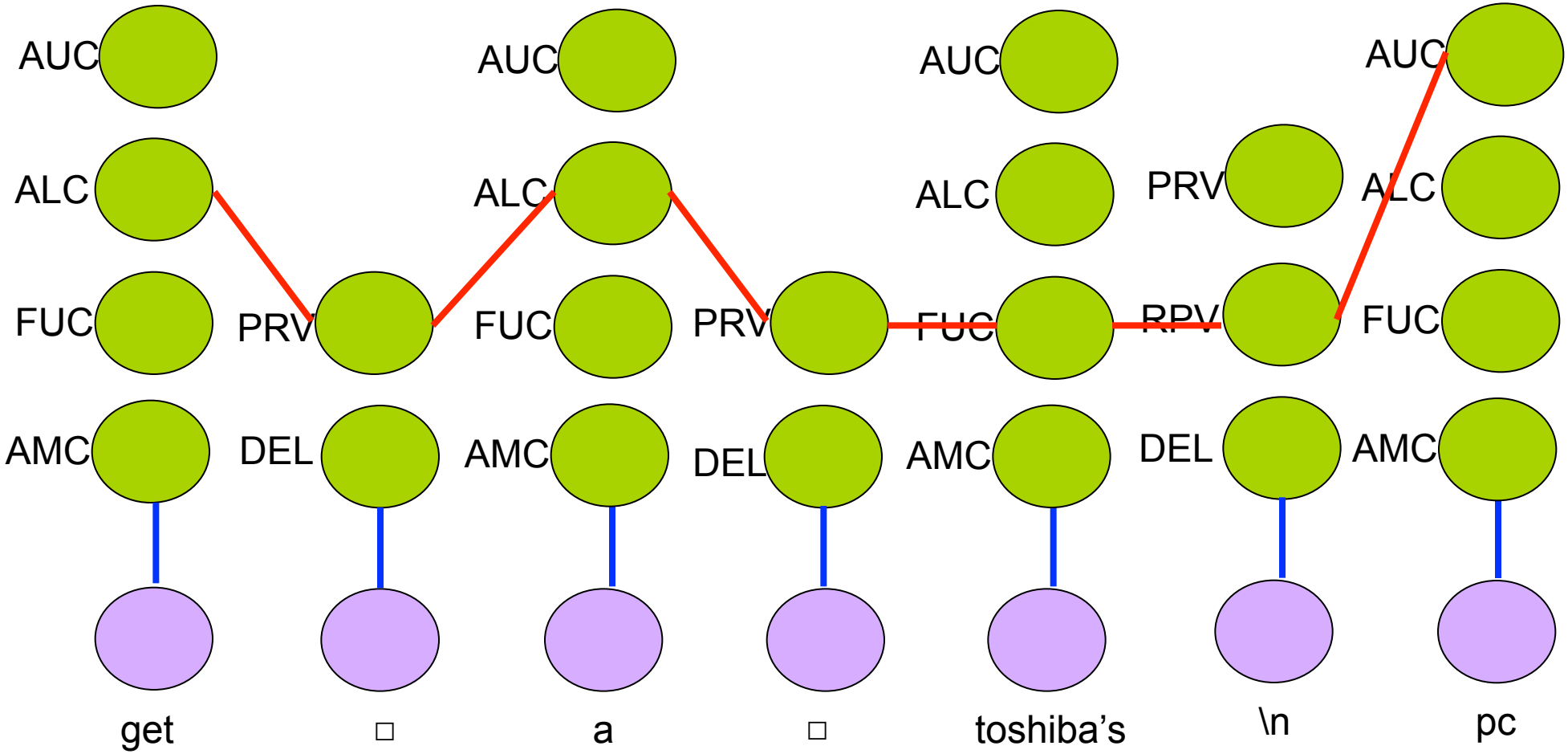
Possible Tags Assignment

- Green nodes are ***tags***
- Purple nodes are ***tokens***



Tagging

$Y^* = \max_Y P(Y|X)$, where X – tokens, Y – tags



Features

Transition Features			
$y_{i-1}=y'$, $y_i=y$			
$y_{i-1}=y'$, $y_i=y$, $w_i=w$			
$y_{i-1}=y'$, $y_i=y$, $t_i=t$			
State Features			
$w_i=w$, $y_i=y$	$w_{i+2}=w$, $y_i=y$	$t_i=t$, $y_i=y$	$t_{i+4}=t$, $y_i=y$
$w_{i-1}=w$, $y_i=y$	$w_{i+3}=w$, $y_i=y$	$t_{i-1}=t$, $y_i=y$	$t_{i-2}=t''$, $t_{i-1}=t'$, $y_i=y$
$w_{i-2}=w$, $y_i=y$	$w_{i+4}=w$, $y_i=y$	$t_{i-2}=t$, $y_i=y$	$t_{i-1}=t'$, $t_i=t$, $y_i=y$
$w_{i-3}=w$, $y_i=y$	$w_{i-1}=w'$, $w_i=w$, $y_i=y$	$t_{i-3}=t$, $y_i=y$	$t_i=t$, $t_{i+1}=t'$, $y_i=y$
$w_{i-4}=w$, $y_i=y$	$w_{i+1}=w'$, $w_i=w$, $y_i=y$	$t_{i-4}=t$, $y_i=y$	$t_{i+1}=t'$, $t_{i+2}=t''$, $y_i=y$
$w_{i+1}=w$, $y_i=y$		$t_{i+1}=t$, $y_i=y$	$t_{i-2}=t''$, $t_{i-1}=t'$, $t_i=t$, $y_i=y$
		$t_{i+2}=t$, $y_i=y$	$t_{i-1}=t''$, $t_i=t$, $t_{i+1}=t'$, $y_i=y$
		$t_{i+3}=t$, $y_i=y$	$t_i=t$, $t_{i+1}=t'$, $t_{i+2}=t''$, $y_i=y$

In total, more than 4M features were used in our experiments

Outline

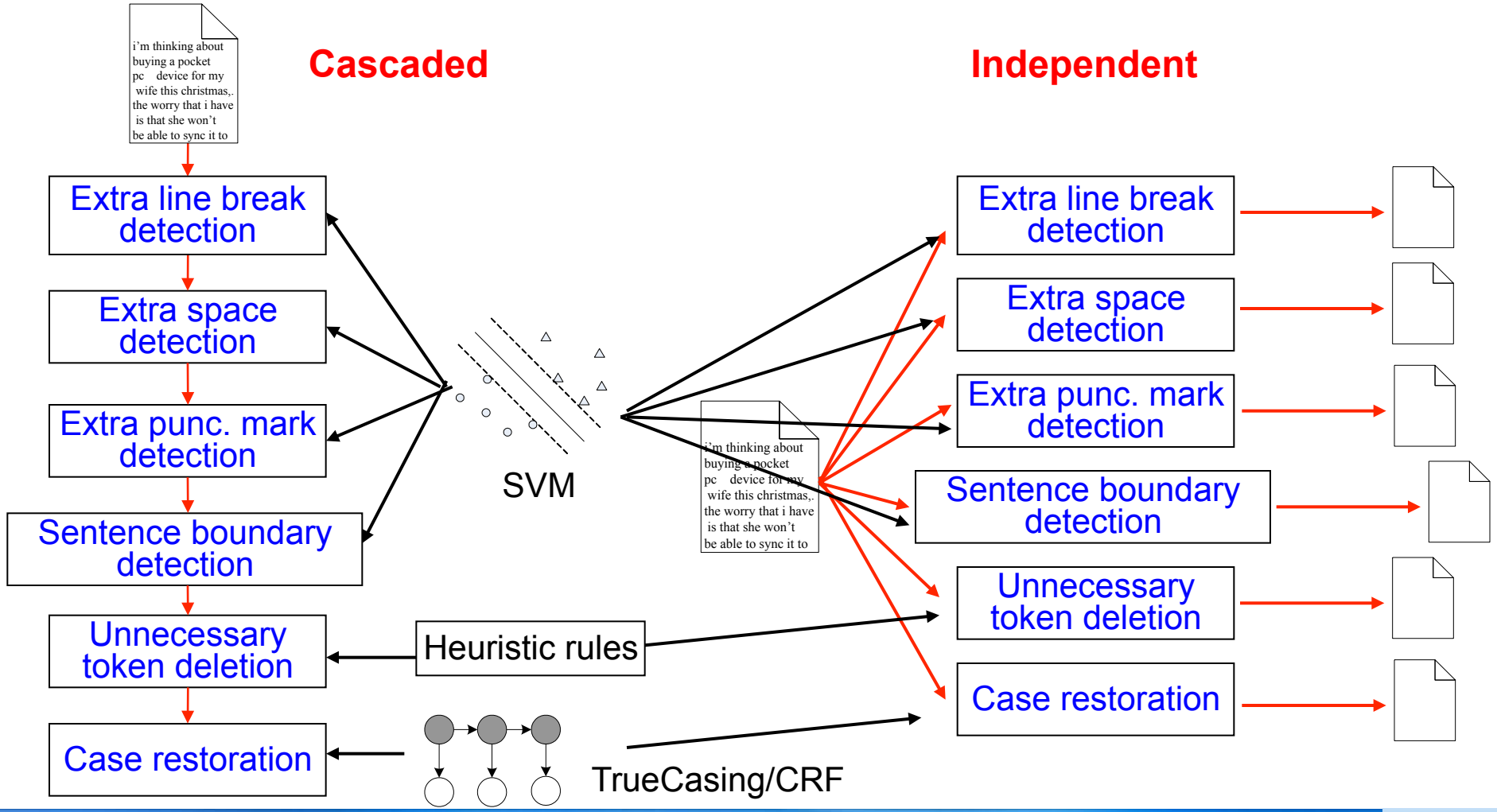
- Motivation
- Related Work
- Problem Description
- A Unified Tagging Approach
- Experimental Results
- Summary

Datasets in Experiments

Data Set	Number of Email	Number of Noises	Extra Line Break	Extra Space	Extra Punc.	Missing Space	Missing Punc.	Casing Error	Spelling Error	Misused Punc.	Unnecessary Token	Number of Paragraph Boundary	Number of Sentence Boundary
DC	100	702	476	31	8	3	24	53	14	2	91	457	291
Ontology	100	2,731	2,132	24	3	10	68	205	79	15	195	677	1,132
NLP	60	861	623	12	1	3	23	135	13	2	49	244	296
ML	40	980	868	17	0	2	13	12	7	0	61	240	589
Jena	700	5,833	3,066	117	42	38	234	888	288	59	1,101	2,999	1,836
Weka	200	1,721	886	44	0	30	37	295	77	13	339	699	602
Protégé	700	3,306	1,770	127	48	151	136	552	116	9	397	1,645	1,035
OWL	300	1,232	680	43	24	47	41	152	44	3	198	578	424
Mobility	400	2,296	1,292	64	22	35	87	495	92	8	201	891	892
WinServer	400	3,487	2,029	59	26	57	142	822	121	21	210	1,232	1,151
Windows	1,000	9,293	3,416	3,056	60	116	348	1,309	291	67	630	3,581	2,742
PSS	1,000	8,965	3,348	2,880	59	153	296	1,331	276	66	556	3,411	2,590
Total	5,000	**41,407**	20,586	6,474	293	645	1,449	6,249	1,418	265	4,028	16,654	13,580

Baseline Methods

Two baselines: cascaded and independent methods



Normalization Results—5-fold cross validation

Detection Task		Prec.	Rec.	F1-measure	Acc.
Extra Line Break	Independent	95.16	91.52	93.30	93.81
	Cascaded	95.16	91.52	93.30	93.81
	Unified	93.87	93.63	**93.75**	**94.53**
Extra Space	Independent	91.85	94.64	93.22	99.87
	Cascaded	94.54	94.56	94.55	99.89
	Unified	95.17	93.98	**94.57**	**99.90**
Extra Punctuation Mark	Independent	88.63	82.69	85.56	99.66
	Cascaded	87.17	85.37	86.26	99.66
	Unified	90.94	84.84	**87.78**	**99.71**
Sentence Boundary	Independent	98.46	99.62	99.04	98.36
	Cascaded	98.55	99.20	98.87	98.08
	Unified	98.76	99.61	**99.18**	**98.61**
Unnecessary Token	Independent	72.51	100.0	84.06	84.27
	Cascaded	72.51	100.0	84.06	84.27
	Unified	98.06	95.47	**96.75**	**96.18**
Case Restoration (TrueCasing)	Independent	27.32	87.44	41.63	96.22
	Cascaded	28.04	88.21	42.55	96.35
Case Restoration (CRF)	Independent	84.96	62.79	72.21	99.01
	Cascaded	85.85	63.99	73.33	99.07
	Unified	86.65	67.09	**75.63**	**99.21**

Normalization Results (cont.)

Text Normalization	Prec.	Rec.	F1	Acc.
Independent (TrueCasing)	69.54	91.33	78.96	97.90
Independent (CRF)	85.05	92.52	88.63	98.91
Cascaded (TrueCasing)	70.29	92.07	79.72	97.88
Cascaded (CRF)	85.06	92.70	88.72	98.92
Unified w/o Transition Features	86.03	93.45	89.59	99.01
Unified	**86.46**	**93.92**	**90.04**	**99.05**

1) The baseline methods suffered from ignorance of the dependencies between the subtasks
2) Our method benefits from modeling the dependencies

Comparison Example

I'm thinking about buying a pocket PC __ device for my wife this Christmas, The worry that I have is that she won't be able to sync it to her outlook express contacts.//

I'm thinking about buying a pocket PC device for my wife this Christmas, the worry that I have is that she won't be able to sync it to her outlook express contacts.//

By independent method

By cascaded method

Original informal text

1. i'm thinking about buying a pocket
2. pc __ device for my wife this christmas,.
3. the worry that i have is that she won't
4. be able to sync it to her outlook express
5. contacts...

By our method

I'm thinking about buying a Pocket PC device for my wife this Christmas.// The worry that I have is that she won't be able to sync it to her Outlook Express contacts.//

Error Analysis

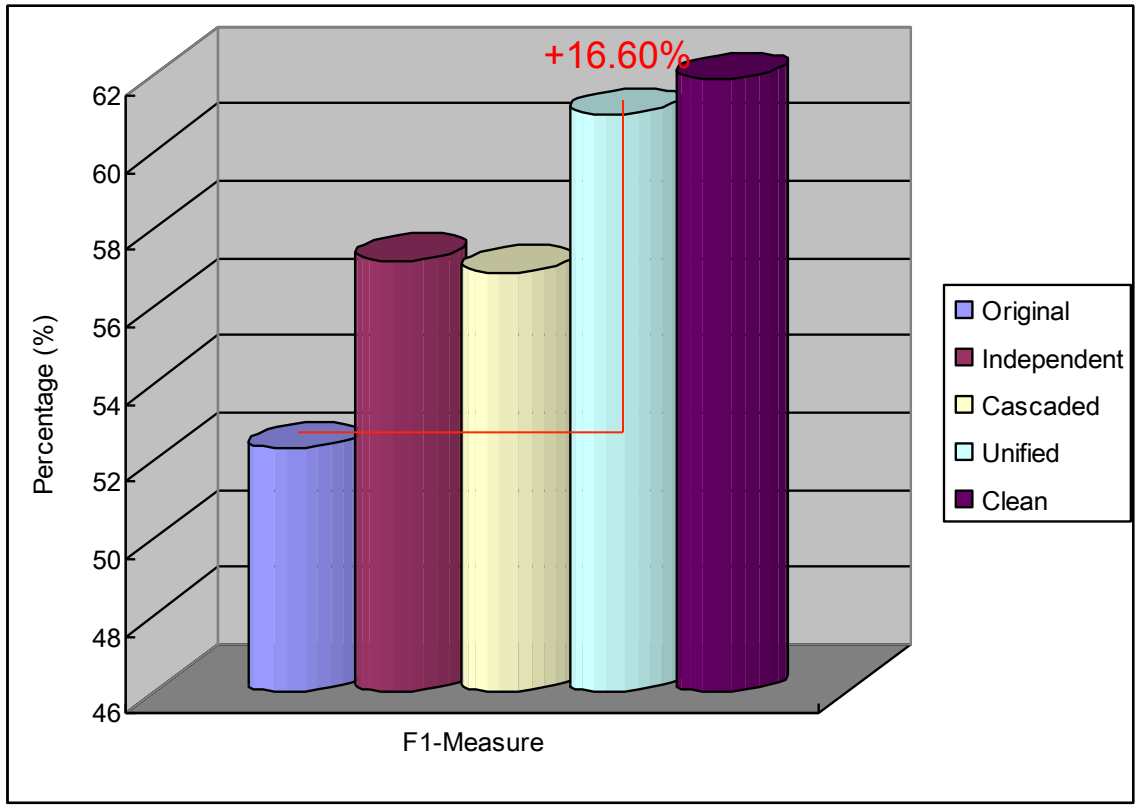
- Extra line break detection
  - 31.14% due to incorrect elimination and 64.07% due to overlooking extra line breaks
- Space detection
  - e.g. “02-16- 2006” and “desk top”
- Case restoration
  - e.g. special word “.NET” and “Ph.D.” and Proper nouns like “John” and “HP Compaq”

Computational Cost

Methods	Training	Tagging
Independent (TrueCasing)	2 minutes	a few seconds
Cascaded (TrueCasing)	3 minutes	a few seconds
Unified	5 hours	25s

*Tested on a computer with two 2.8G P4-CPUs and 3G memory

How Text Normalization Helps NER



Outline

- Motivation
- Related Work
- Problem Description
- A Unified Tagging Approach
- Experimental Results
- Summary

Summary

- Investigated the problem of text normalization
- Formalized the problem as a task of noise elimination and boundary detection subtasks
- Proposed a unified tagging approach to perform the subtasks together
- Empirical verification of the effectiveness of the proposed approach

Thanks!

HP: http://keg.cs.tsinghua.edu.cn/~jietang/

Email: jietang@tsinghua.edu.cn