Information Extraction

(Part 2)

Shobeir Fakhraei
University of Southern California

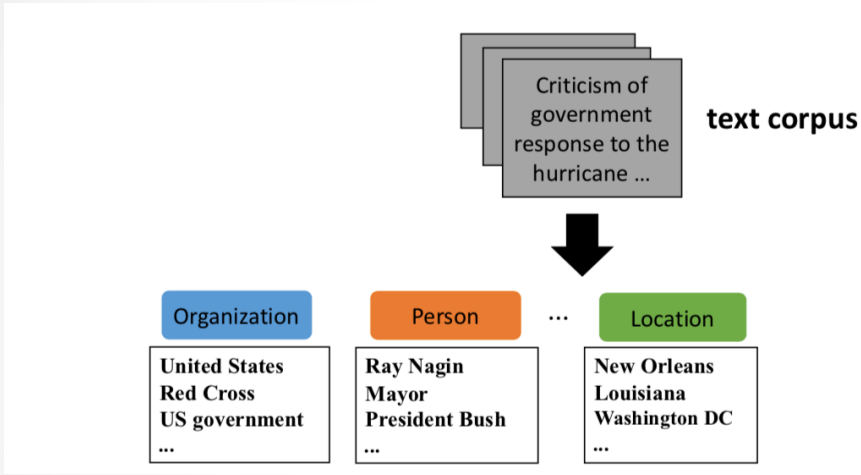
Slides based on material provided by Andrew McCallum, William Cohen, Matt Michelson, Xiang Ren, and others.

Where are we?

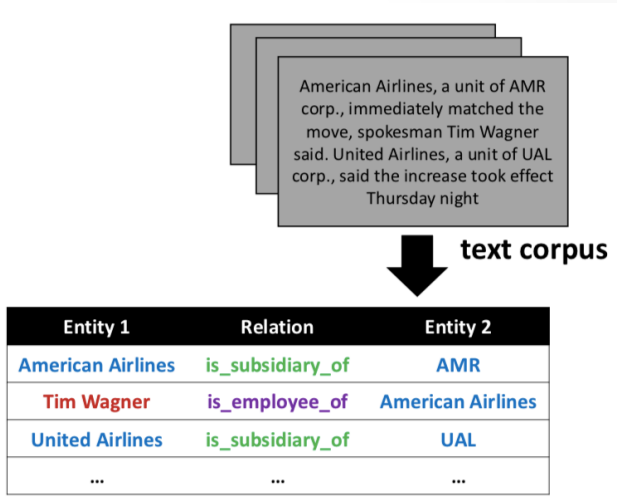
- Information Extraction
  - Semi-structured: Web, HTML
  - **Unstructured: Text, Adds, Tweets, …**
- Entity Linkage, Data Cleaning, Normalization
- Logical Data Integration
  - Mediators, Query Rewriting
  - Warehouse, Logical Data Exchange
- Automatic Source Modeling/Learning Schema Mappings
- Semantic Web
  - RDF, SPARQL, OWL, Linked Data
- Advanced Topics
  - Geospatial Data Integration, Knowledge Graphs

Landscape of IE Tasks (1): Types of Extractions

Named Entity Recognition

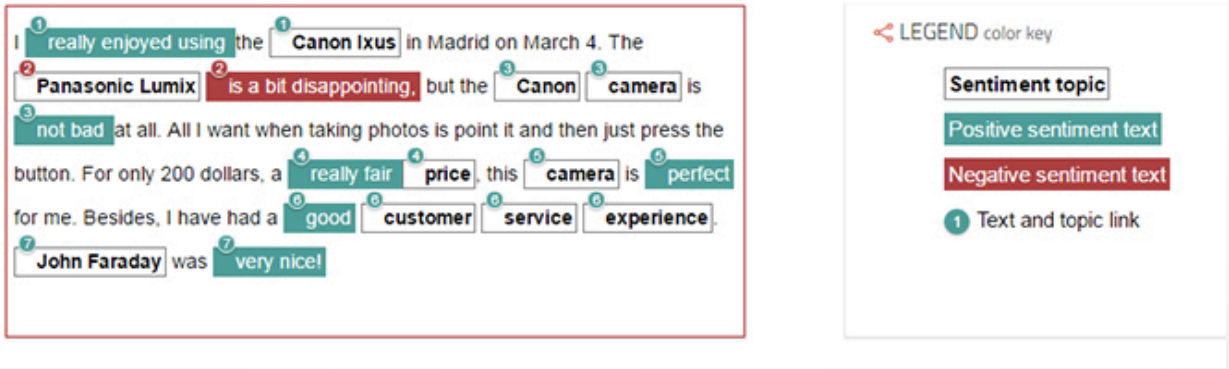


Relationship Extraction



Entity 1	Relation	Entity 2
American Airlines	is_subsidiary_of	AMR
Tim Wagner	is_employee_of	American Airlines
United Airlines	is_subsidiary_of	UAL
...	...	...

Adjective Extractions -> Sentiment Analysis



Entity Recognition (and Typing)

- **Identify token spans of entity mentions in text, and classify them into types of interest**

  [*Barack Obama*] *arrived this afternoon in* [*Washington, D.C.*].
  [*President Obama*]*'s wife* [*Michelle*] *accompanied him*

  [*TNF alpha*] *is produced chiefly by activated* [*macrophages*]

NER Approaches

1. Rule/Pattern-Based
2. Standard Classifiers
   - KNN, decision tree, naïve Bayes, SVM, ...
3. Sequence Models
   - HMMs, CRFs, LSTM-CRF

Workflow of Token-wise Classifiers

Training

1. Collect a set of representative training documents
2. Label each token for its entity class or other (O)
3. Design feature extractors appropriate to the text and classes
4. Train a classifier to predict the labels of each token in the annotated training sentences

Testing

1. Receive a set of testing documents
2. Run trained classifier to label each token
3. Appropriately output the recognized entities

Token-Level Input/Label

Input: X = a sequence of tokens
Label: Y = a sequence of labels e.g.
Y ={HouseNo, Street, City, State, Zip, Country, Other}.

Here is my review of Fermat's last theorem by S. Singh

i	1	2	3	4	5	6	7	8	9	10	11
x	Here	is	my	review	of	Fermat's	last	theorem	by	S.	Singh
y	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}

R. Fagin and J. Helpbern, Belief Awareness Reasoning

i	1	2	3	4	5	6	7	8	9
x	R.	Fagin	and	J.	Helpbern	,	Belief	Awareness	Reasoning
y	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9

Multi-token Entity labeling Schemas

BCEO (B=Begin, C=Continue, E=End, O=Other)
BIO (B=Begin, I=Inside, O=Other)

Features for NER

- **Words**
  - Current word (essentially like a learned dictionary)
  - Previous/next word (context)
- **Orthographic**
  - E.g., Capitalizations, presence of dot
- **Other kinds of inferred linguistic classification**
  - Part-of-speech tags
- **Label context**
  - Previous (and perhaps next) label

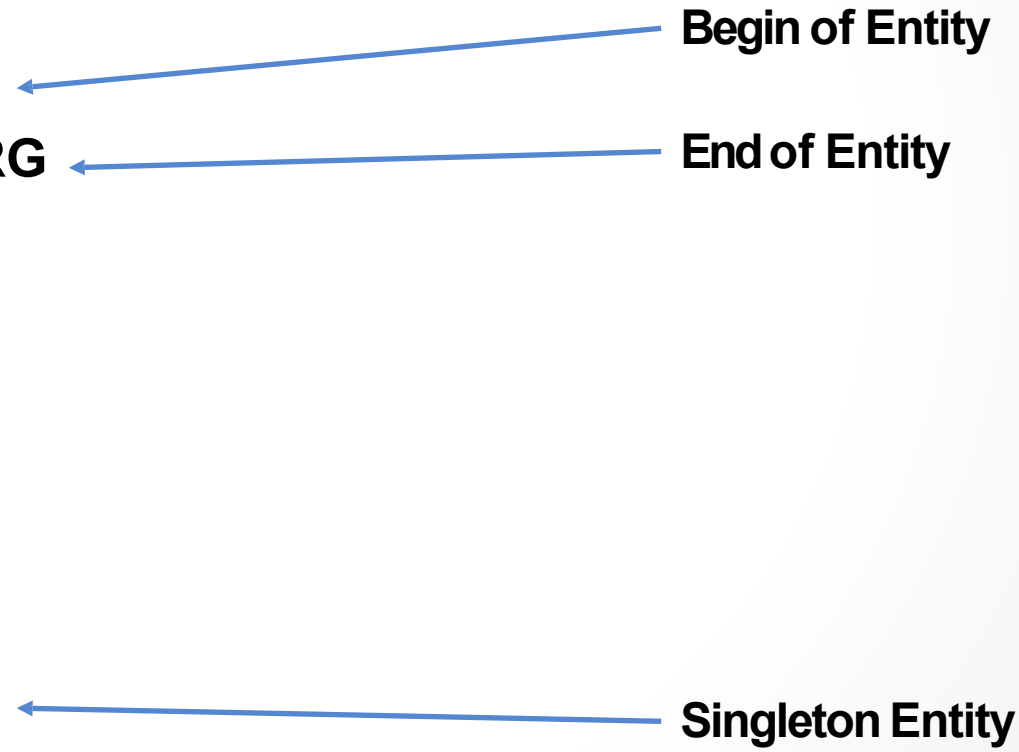
Classical Learning Models on NER

- **KNN**
- **Decision Tree**
- **Naïve Bayes**
- **SVM**
- **Boosting, …**

NER as Sequence Labeling

Encoding labels for sequence

- BIOES labeling schema:
  - The O
  - European B-ORG
  - Commission E-ORG
  - said O
  - on O
  - Thursday O
  - it O
  - disagreed O
  - with O
  - German S-MISC



Sequence Labeling vs. Classification

- **Sequence Models are statistical models of *multiple* token sequences that effectively label sub-sequences**
- **The goal is to incorporate context**

Sequence problems

- Many problems in NLP have data which is a sequence of characters, words, phrases, lines, or sentences …
- We can think of our task as one of labeling each item

VBG	NN	IN	DT	NN	IN	NN
Chasing	opportunity	in	an	age	of	upheaval

POS tagging

B	B	I	I	B	I	B	I	B	B
而	相	对	于	这	些	品	牌	的	价

Word segmentation

PERS	O	O	O	ORG	ORG
Murdoch	discusses	future	of	News	Corp.

Named entity recognition

Text segmentation

MEMM inference in systems

- Maximum Entropy Markov Model (MEMM): The classifier makes a single decision at a time, conditioned on evidence from observations and previous decisions
- A larger space of sequences is usually explored via search

Local Context

Decision Point

-3	-2	-1	0	+1
DT	NNP	VBD	???	???
The	Dow	fell	22.6	%

Features

W_0	22.6
W_{+1}	%
W_{-1}	fell
T_{-1}	VBD
T_{-1}-T_{-2}	NNP-VBD
hasDigit?	true
…	…

(Ratnaparkhi 1996; Toutanova et al. 2003, etc.)

Example: POS Tagging

- Scoring individual labeling decisions is no more complex than standard classification decisions
  - We have some assumed labels to use for prior positions
  - We use features of those and the observed data (which can include current, previous, and next words) to predict the current label

Local Context

Decision Point

-3	-2	-1	0	+1
DT	NNP	VBD	???	???
The	Dow	fell	22.6	%

Features

W_0	22.6
W_{+1}	%
W_{-1}	fell
T_{-1}	VBD
T_{-1}-T_{-2}	NNP-VBD
hasDigit?	true
…	…

(Ratnaparkhi 1996; Toutanova et al. 2003, etc.)

Example: POS Tagging

- POS tagging Features can include:
  - Current, previous, next words in isolation or together.
  - Previous one, two, three tags.
  - Word-internal features: word types, suffixes, dashes, etc.

Local Context

Decision Point

-3	-2	-1	0	+1
DT	NNP	VBD	???	???
The	Dow	fell	22.6	%

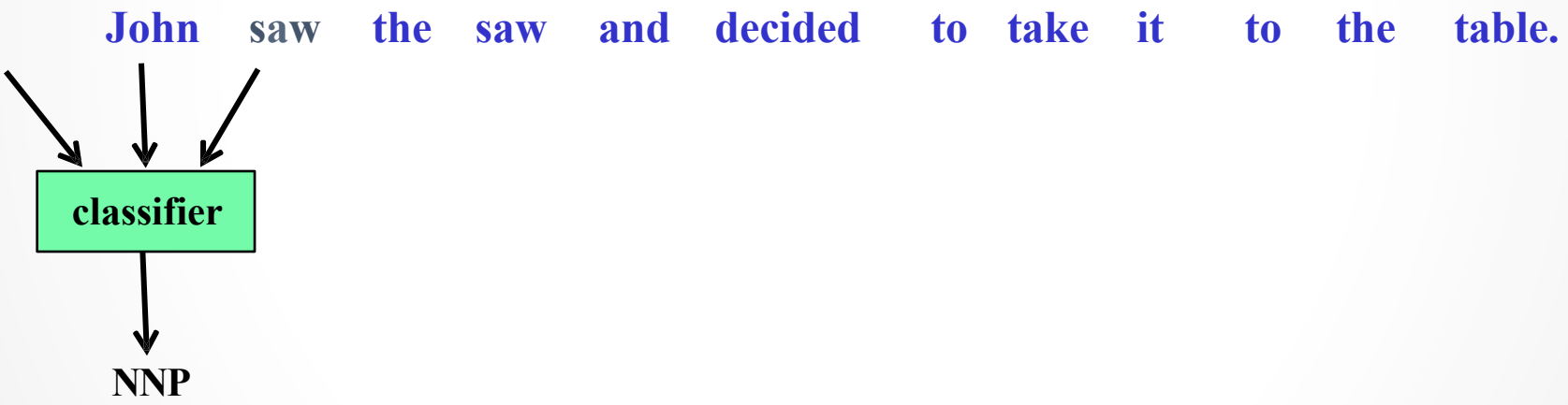
Features

W_0	22.6
W_{+1}	%
W_{-1}	fell
T_{-1}	VBD
T_{-1}-T_{-2}	NNP-VBD
hasDigit?	true
…	…

(Ratnaparkhi 1996; Toutanova et al. 2003, etc.)

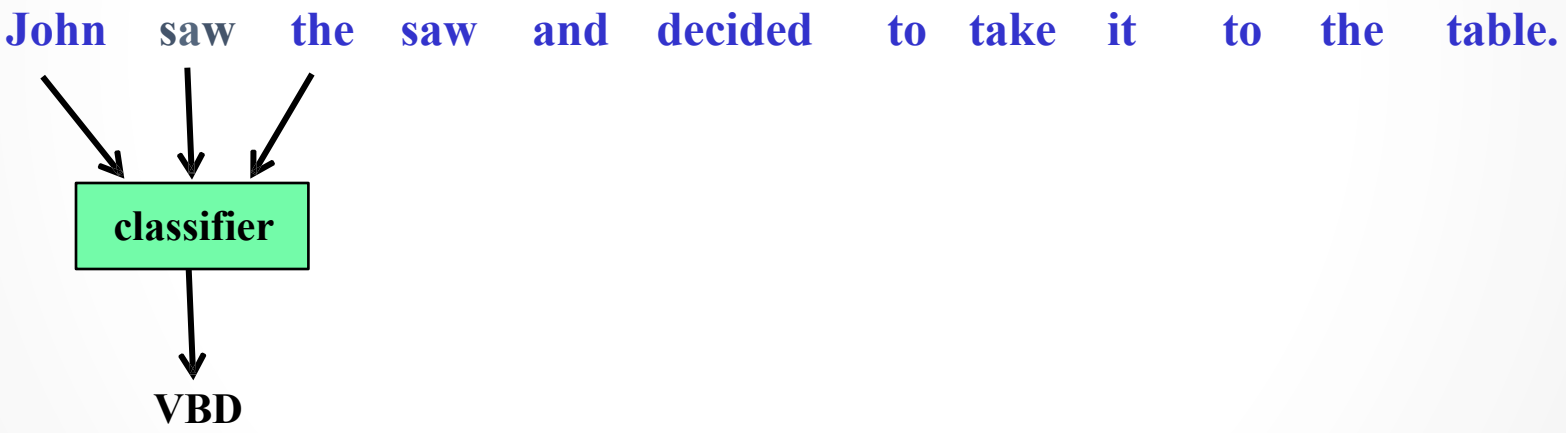
Sequence Labeling as Classification

- **Classify each token *independently* but use as input features, information about the surrounding tokens (sliding window).**



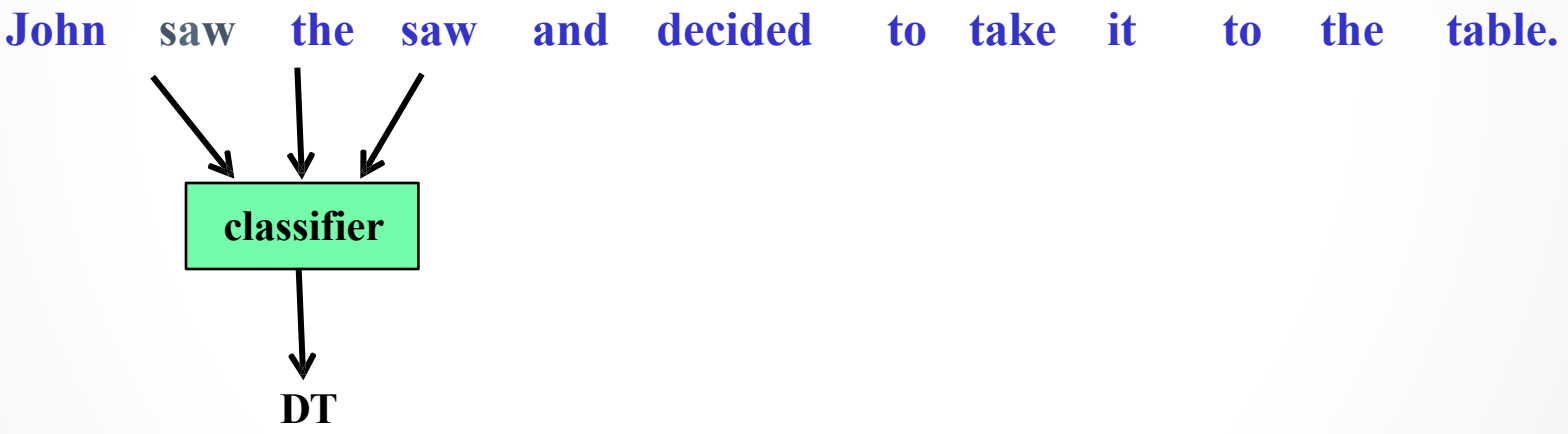
Sequence Labeling as Classification

- **Classify each token *independently* but use as input features, information about the surrounding tokens (sliding window).**



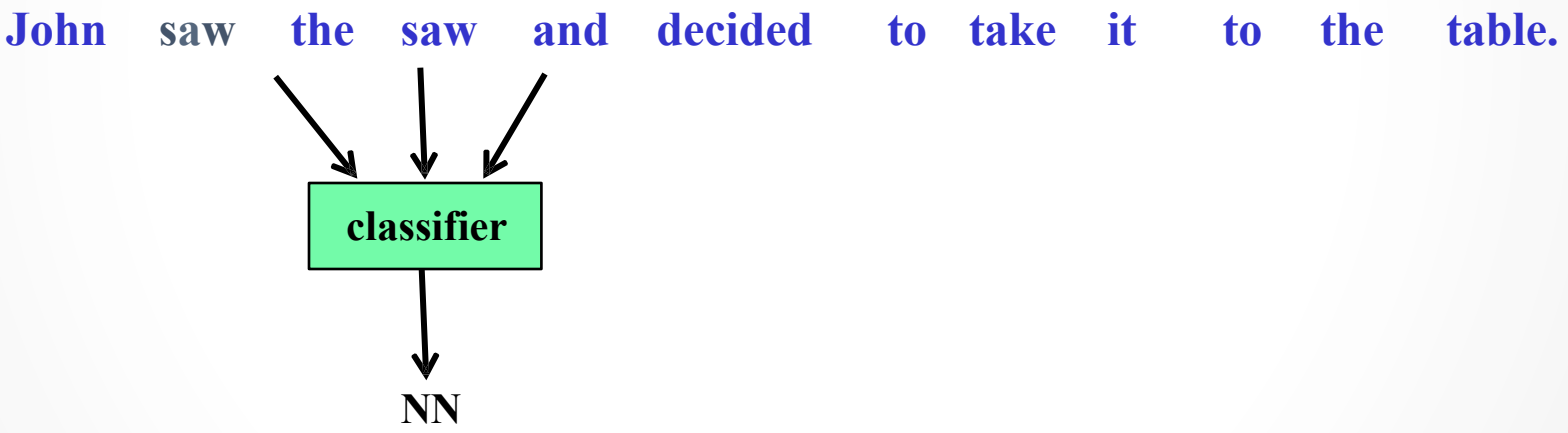
Sequence Labeling as Classification

- **Classify each token *independently* but use as input features, information about the surrounding tokens (sliding window).**



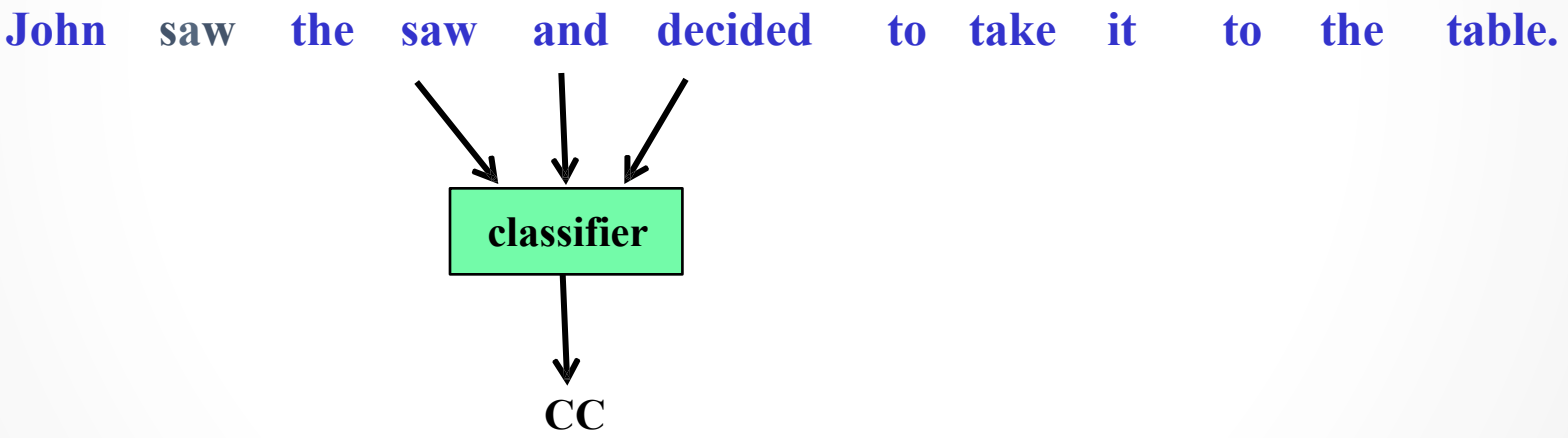
Sequence Labeling as Classification

- **Classify each token *independently* but use as input features, information about the surrounding tokens (sliding window).**



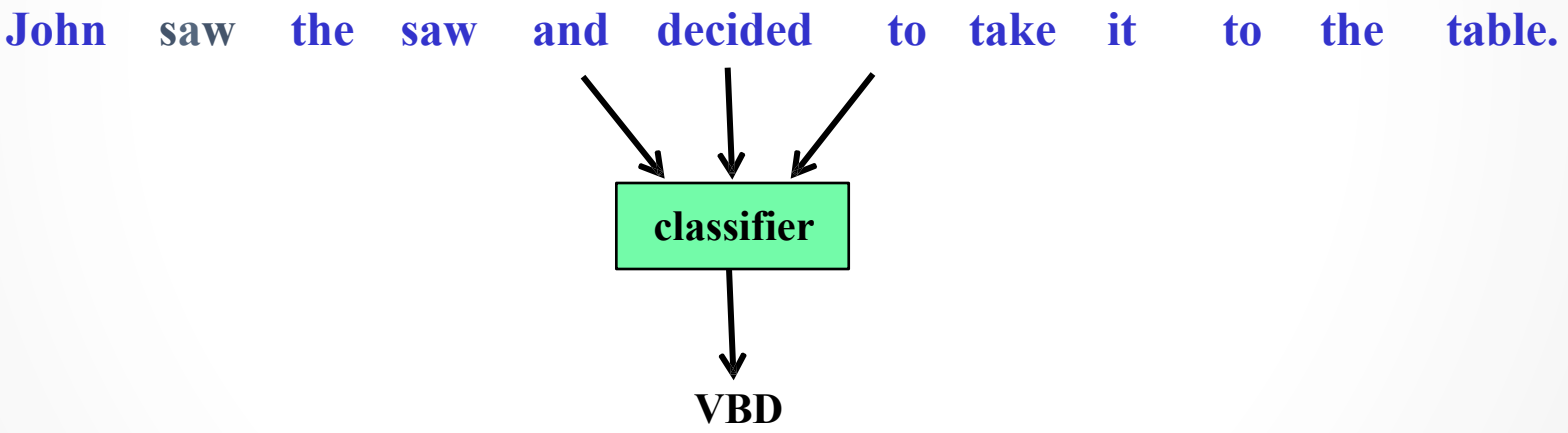
Sequence Labeling as Classification

- **Classify each token *independently* but use as input features, information about the surrounding tokens (sliding window).**



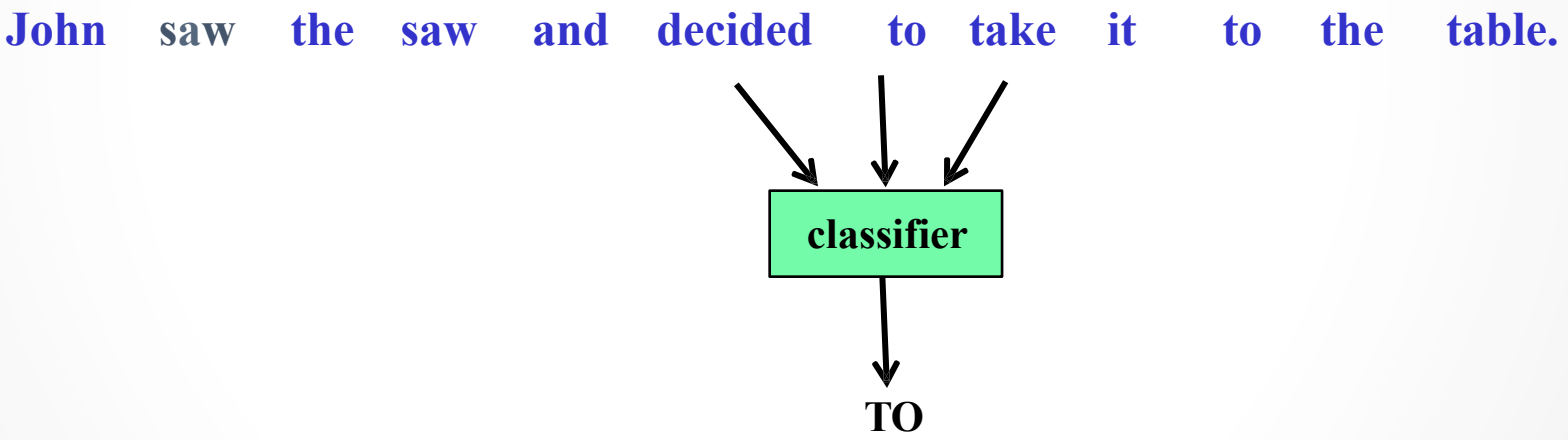
Sequence Labeling as Classification

- **Classify each token *independently* but use as input features, information about the surrounding tokens (sliding window).**



Sequence Labeling as Classification

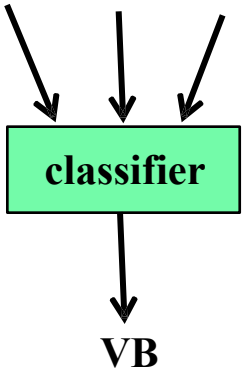
- **Classify each token *independently* but use as input features, information about the surrounding tokens (sliding window).**



Sequence Labeling as Classification

- **Classify each token *independently* but use as input features, information about the surrounding tokens (sliding window).**

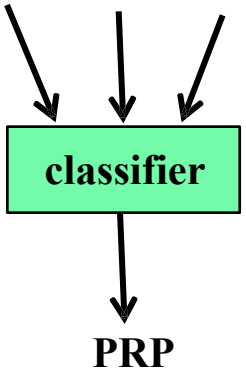
John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

- **Classify each token *independently* but use as input features, information about the surrounding tokens (sliding window).**

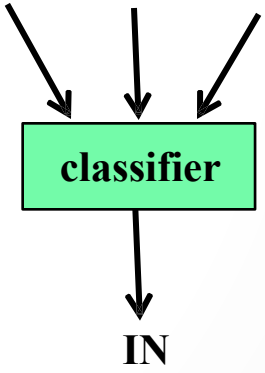
John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

- **Classify each token *independently* but use as input features, information about the surrounding tokens (sliding window).**

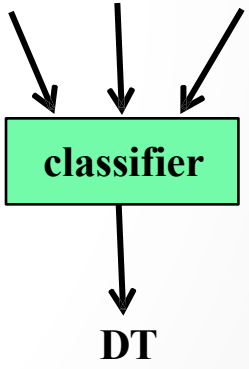
John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

- **Classify each token *independently* but use as input features, information about the surrounding tokens (sliding window).**

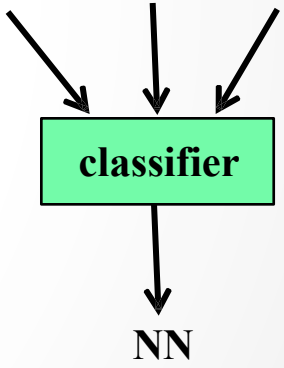
John saw the saw and decided to take it to the table.



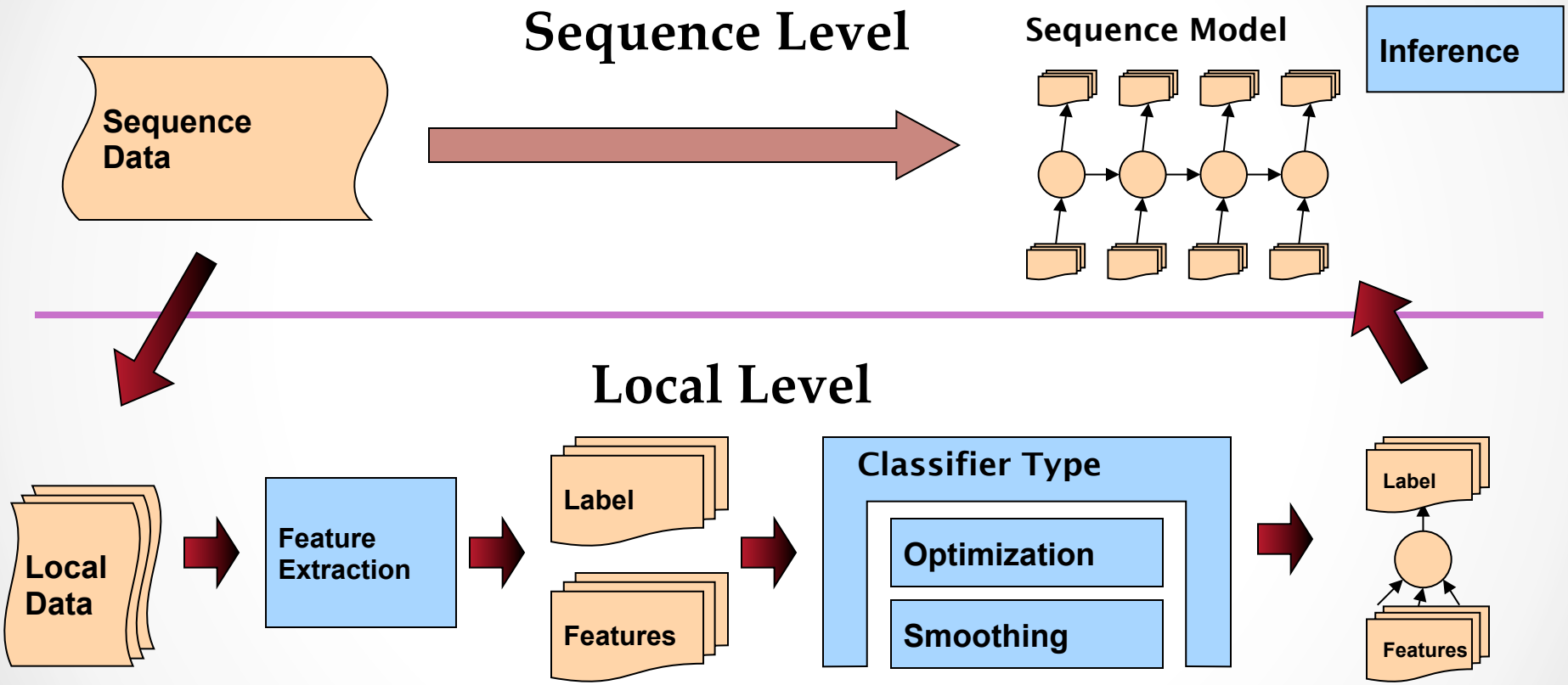
Sequence Labeling as Classification

- **Classify each token *independently* but use as input features, information about the surrounding tokens (sliding window).**

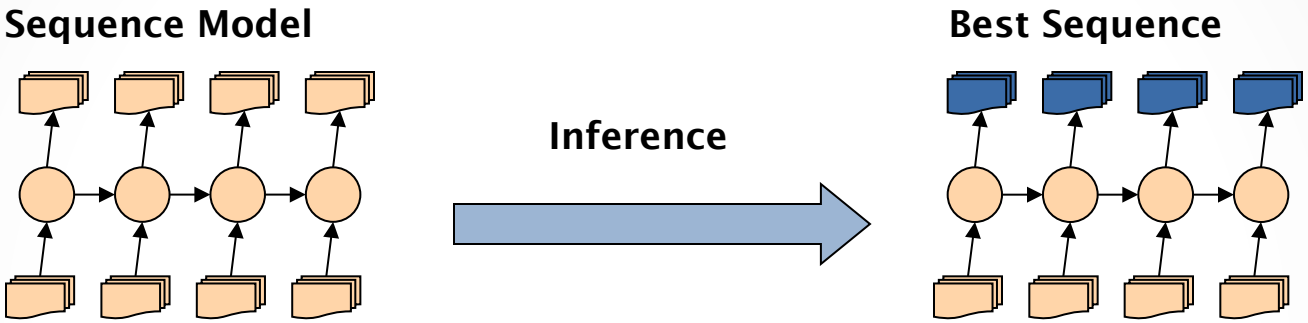
John saw the saw and decided to take it to the table.



Inference in Systems

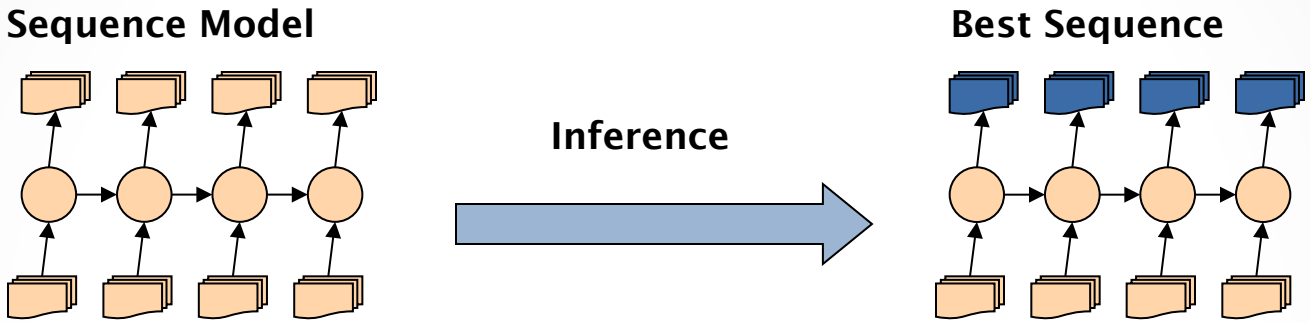


Greedy Inference



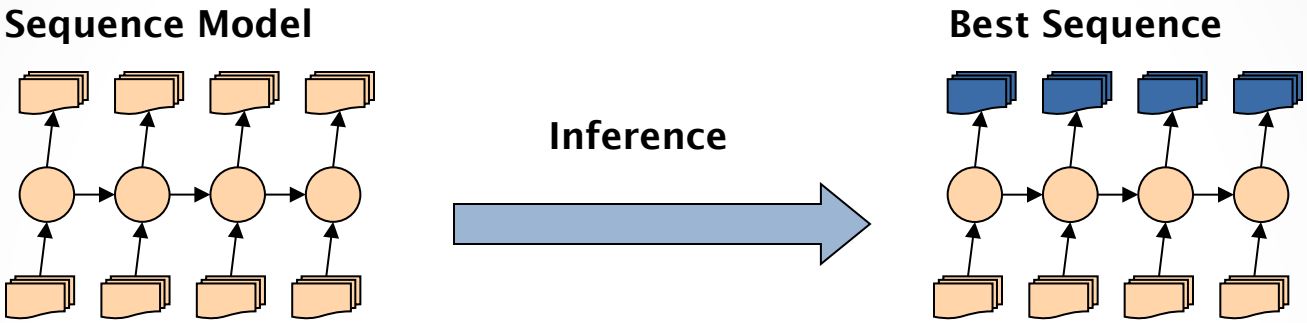
- Greedy inference:
  - We just start at the left, and use our classifier at each position to assign a label
  - The classifier can depend on previous labeling decisions as well as observed data
- Advantages:
  - Fast, no extra memory requirements
  - Very easy to implement
  - With rich features including observations to the right, it may perform quite well
- Disadvantage:
  - Greedy. We make commit errors we cannot recover from

Beam Inference



- Beam inference:
  - At each position keep the top k complete sequences.
  - Extend each sequence in each local way.
  - The extensions compete for the k slots at the next position.
- Advantages:
  - Fast; beam sizes of 3–5 are almost as good as exact inference in many cases.
  - Easy to implement (no dynamic programming required).
- Disadvantage:
  - Inexact: the globally best sequence can fall off the beam.

Viterbi Inference



- Viterbi inference:
  - Dynamic programming or memoization.
  - Requires small window of state influence (e.g., past two states are relevant).
- Advantage:
  - Exact: the global best sequence is returned.
- Disadvantage:
  - Harder to implement long-distance state-state interactions.

Conditional Random Fields

- A whole-sequence conditional model rather than a chaining of local models.

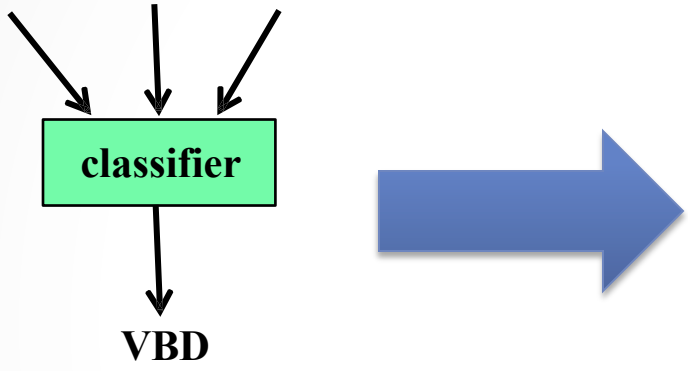
$$P(c \mid d, \lambda)$$

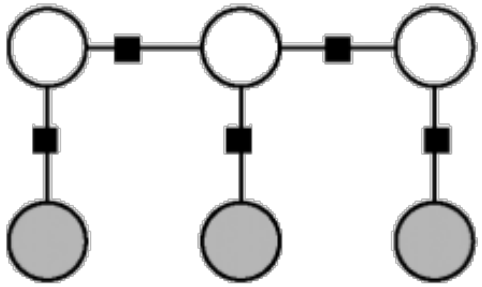
- The space of c' s is now the space of sequences
- Training is slower, but CRFs avoid causal-competition biases
- Are used in the state-of-the-art solutions.

Conditional Random Fields

What is the difference?

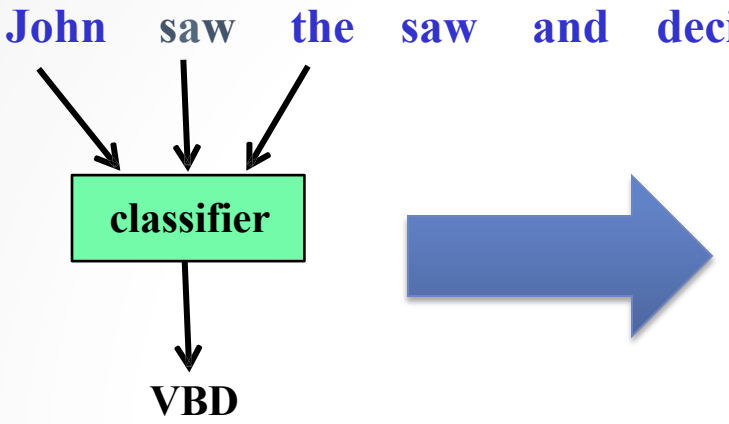
John saw the saw and decided to take it to the table.

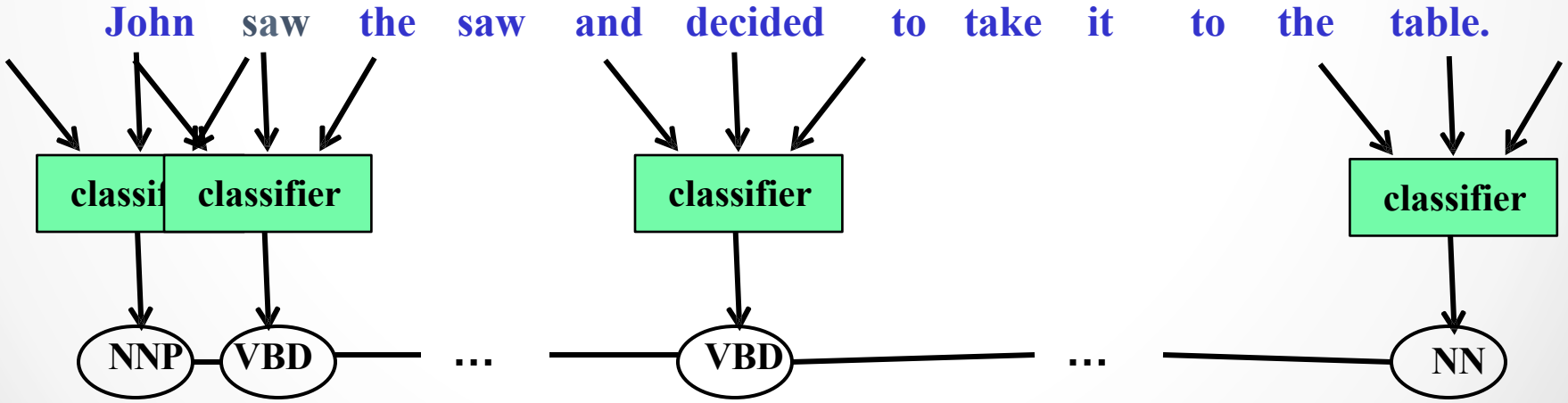




Linear-chain CRFs

What is the difference?





Structured Prediction

Single output prediction:

$$(x_1, .. x_n) \rightarrow y$$

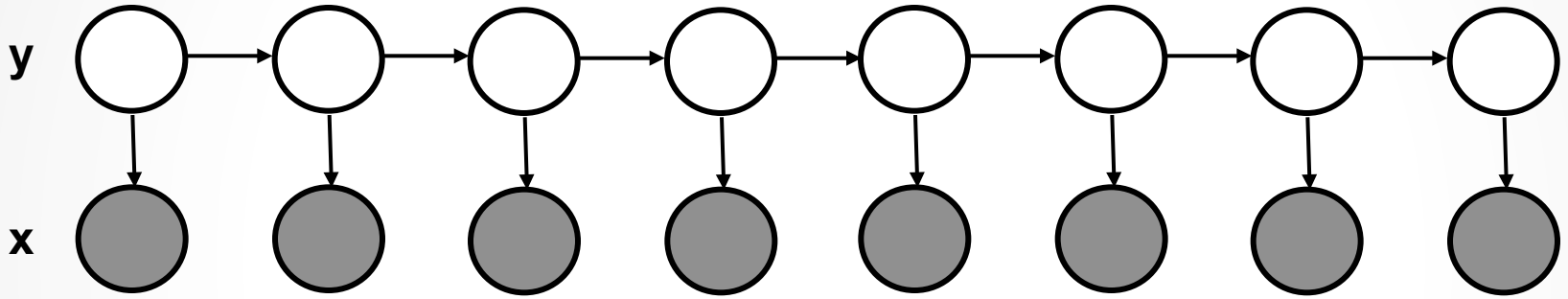
Structured prediction:

$$(x_1, .. x_n) \rightarrow y_1, .. , y_m$$

Here is my review of Fermat's last theorem by S. Singh

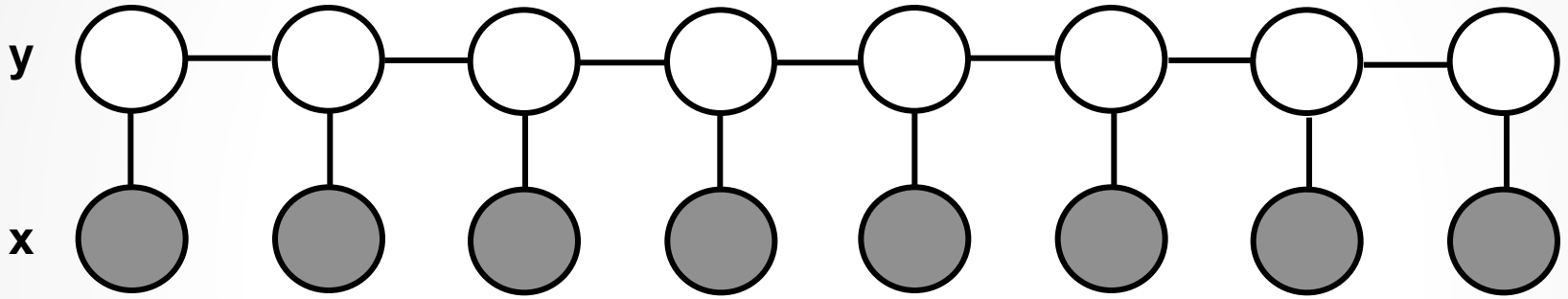
i	1	2	3	4	5	6	7	8	9	10	11
x	Here	is	my	review	of	Fermat's	last	theorem	by	S.	Singh
y	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}

Hidden Markov Models (HMMs)



- Generative
  - Find parameters to maximize P(X,Y)

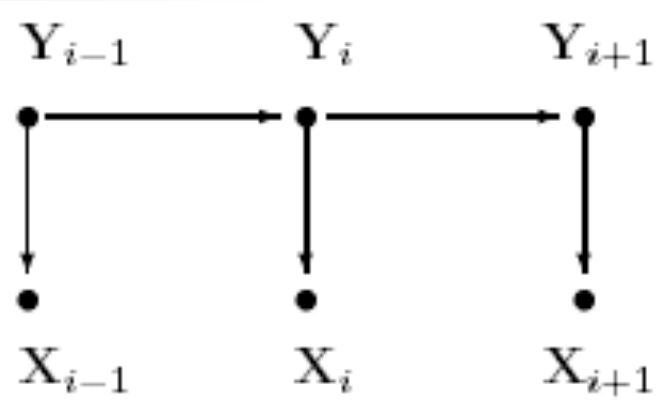
Chain Conditional Random Fields (CRFs)



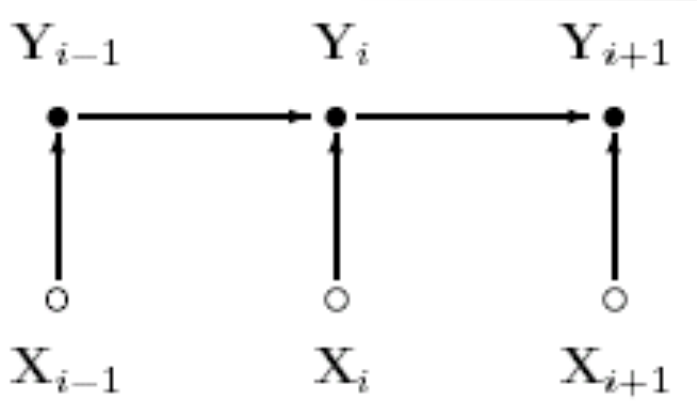
- Discriminative
  - Find parameters to maximize P(Y | X)

Generative vs Discriminative

Generative



Discriminative



- Generative:
  - Assumes a generative process.
    - i.e. Class generates samples
  - Maximize P(y,x) to estimate P(y | x)
- Discriminative:
  - Does not assume generative process
  - Directly maximize P(y | x)
  - No need to approximate p(x)

Background

Bayes Theorem

- Definition of Conditional Probability

$$P(A \mid B) = \frac{P(A,B)}{P(B)}$$

- Bayes Rule

(Thomas Bayes, 1763)

$$P(B \mid A) = \frac{P(A,B)}{P(A)} = \frac{P(A \mid B)P(B)}{P(A)}$$

- Corollary: The Chain Rule

$$P(A \mid B)P(B) = P(A,B)$$

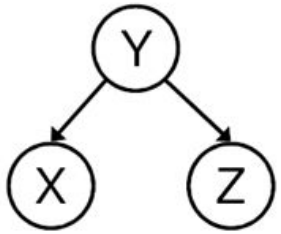
$$P(q_t, q_{t-1}, \ldots q_1) = P(q_t \mid q_{t-1}, \ldots q_1)P(q_{t-1}, \ldots q_1)$$

$$= P(q_t \mid q_{t-1}, \ldots q_1)P(q_{t-1} \mid q_{t-2}, \ldots q_1)P(q_{t-2}, \ldots q_1)$$

$$= P(q_1)\prod_{i=2}^{t} P(q_i \mid q_{i-1}, \ldots q_1)$$

Background
Conditional Independence

- Common cause



Y: Project due

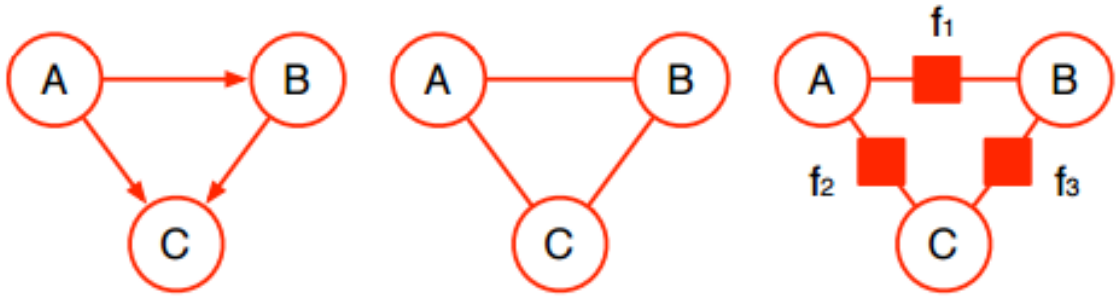
X: Newsgroup busy

Z: Lab full

P(X,Z|Y) = P(X|Y).P(Z|Y)

- Are X and Z independent?
  - No
- Are they conditionally independent given Y?
  - Yes

Background



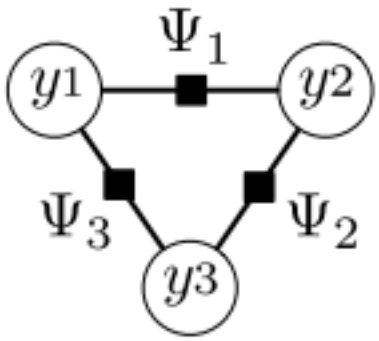
(a) Directed graph (b) Undirected graph (c) Factor graph

- Nodes represent random variables
- Edges reflect dependencies between variables

Factor Graphs

Definition 2.1. A distribution $p(\mathbf{y})$ *factorizes according to a factor graph* G if there exists a set of local functions Ψ_a such that p can be written as

$$p(\mathbf{y}) = Z^{-1} \prod_{a \in F} \Psi_a(\mathbf{y}_{N(a)}) \tag{2.3}$$

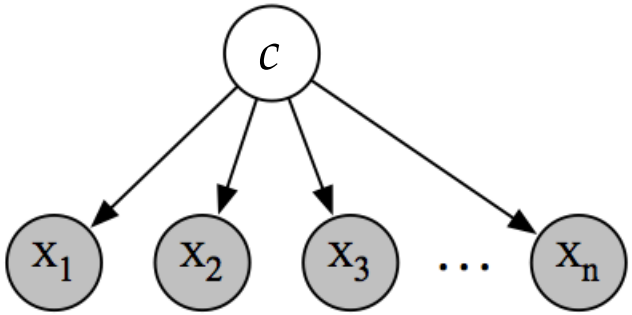


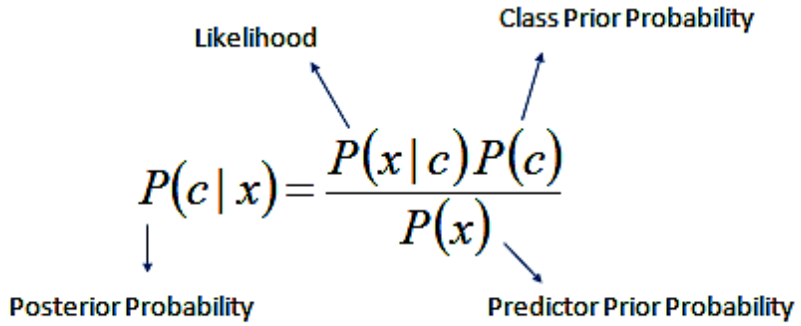
$$p(y_1, y_2, y_3) = \Psi_1(y_1, y_2)\Psi_2(y_2, y_3)\Psi_3(y_1, y_3)$$

For all $\mathbf{y} = (y_1, y_2, y_3)$

Background

- Naïve Bayes



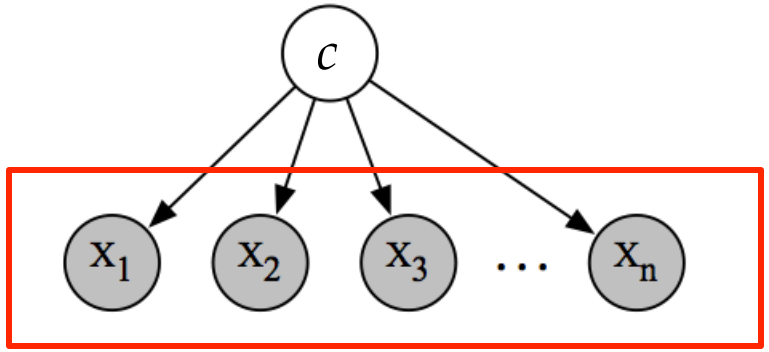


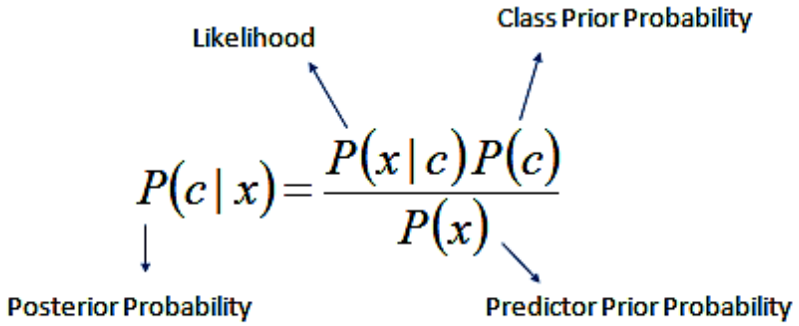
$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Background

- Naïve Bayes

Can be highly correlated





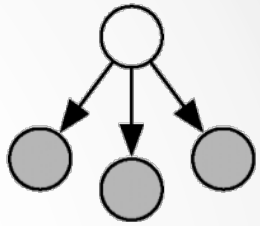
$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Discriminative vs. Generative

$p(\mathbf{y}, \mathbf{x})$

- Generative Model: A model that generate observed data randomly
- Naïve Bayes: once the class label is known, all the features are independent

$$p(y, \mathbf{x}) = p(y) \prod_{k=1}^{K} p(x_k|y)$$



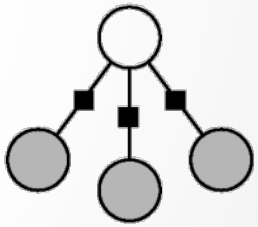
Naive Bayes

CONDITIONAL

$p(\mathbf{y} \mid \mathbf{x})$

- Discriminative: Directly estimate the posterior probability; Aim at modeling the "discrimination" between different outputs
- Logisitic Regression: linear combination of feature function in the exponent,

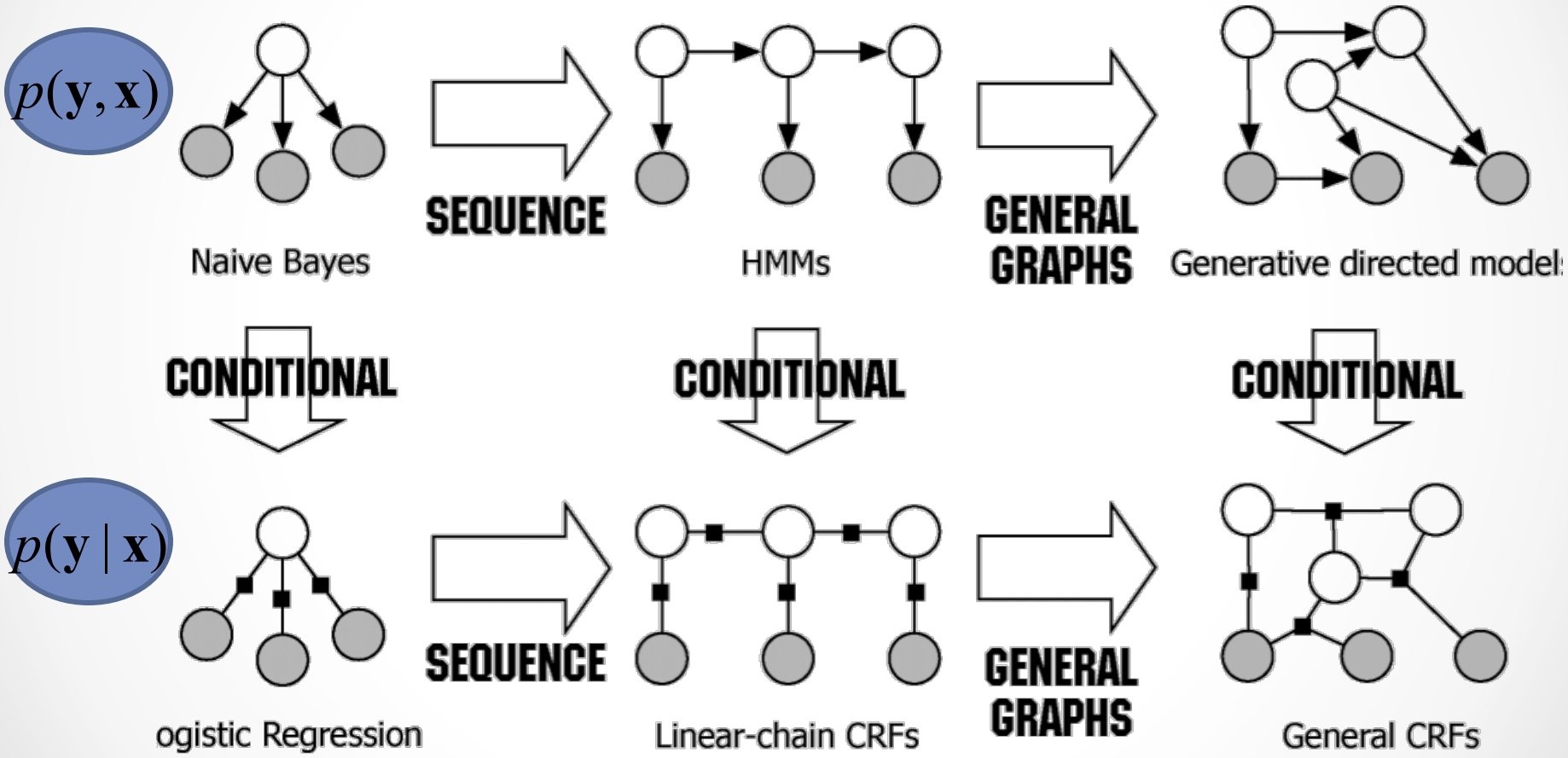
$$p(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left\{\sum_{k=1}^{K} \theta_k f_k(y, \mathbf{x})\right\}$$



Logistic Regression

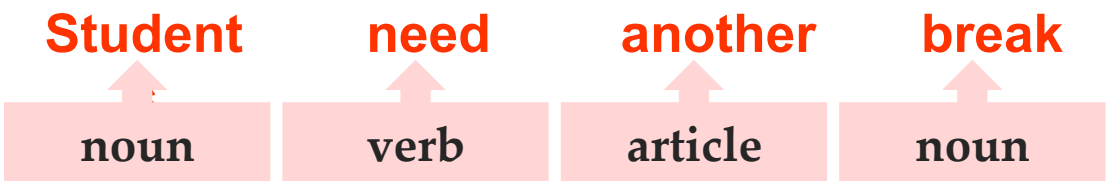
Both generative models and discriminative models describe distributions over (y , x), but they work in different directions.

Generative-Discriminative Pairs

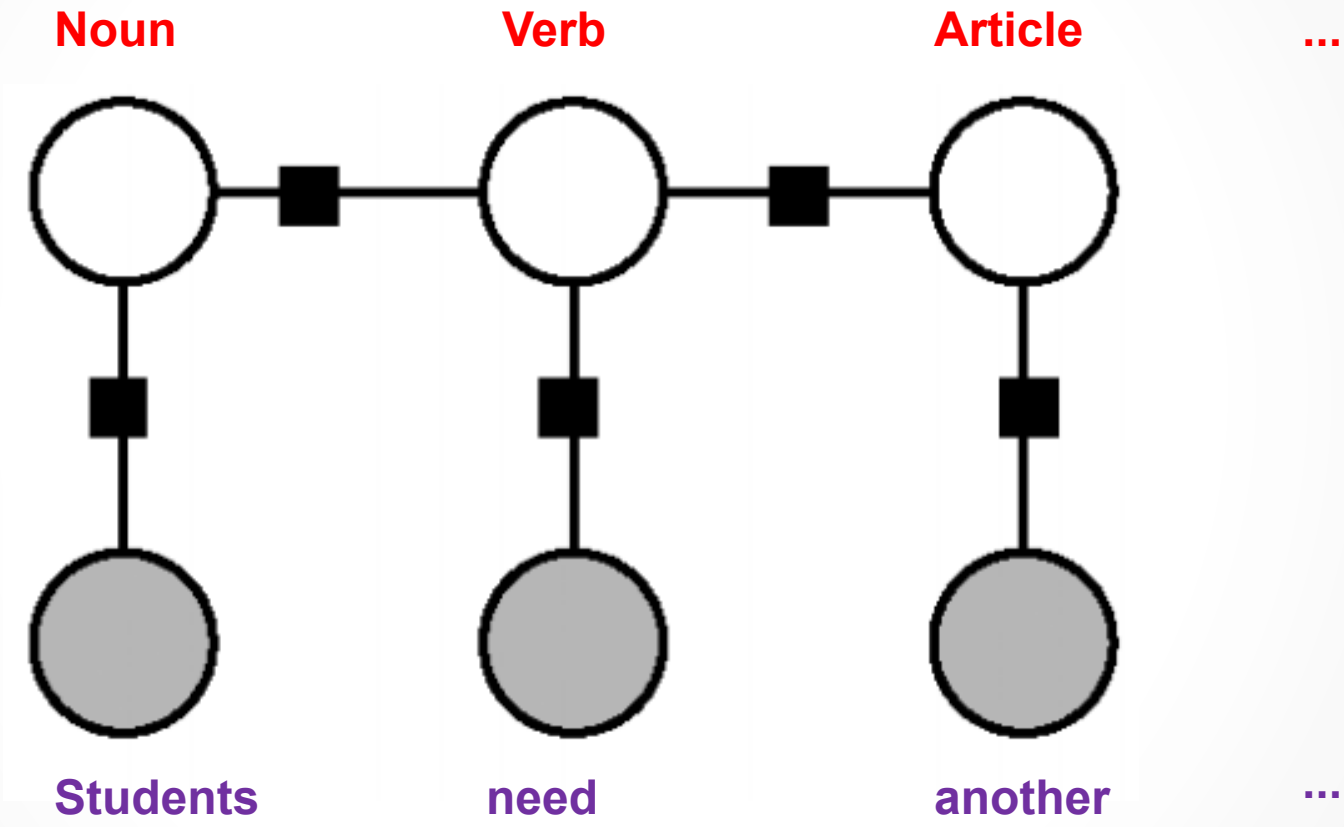


Example: part of speech (POS)

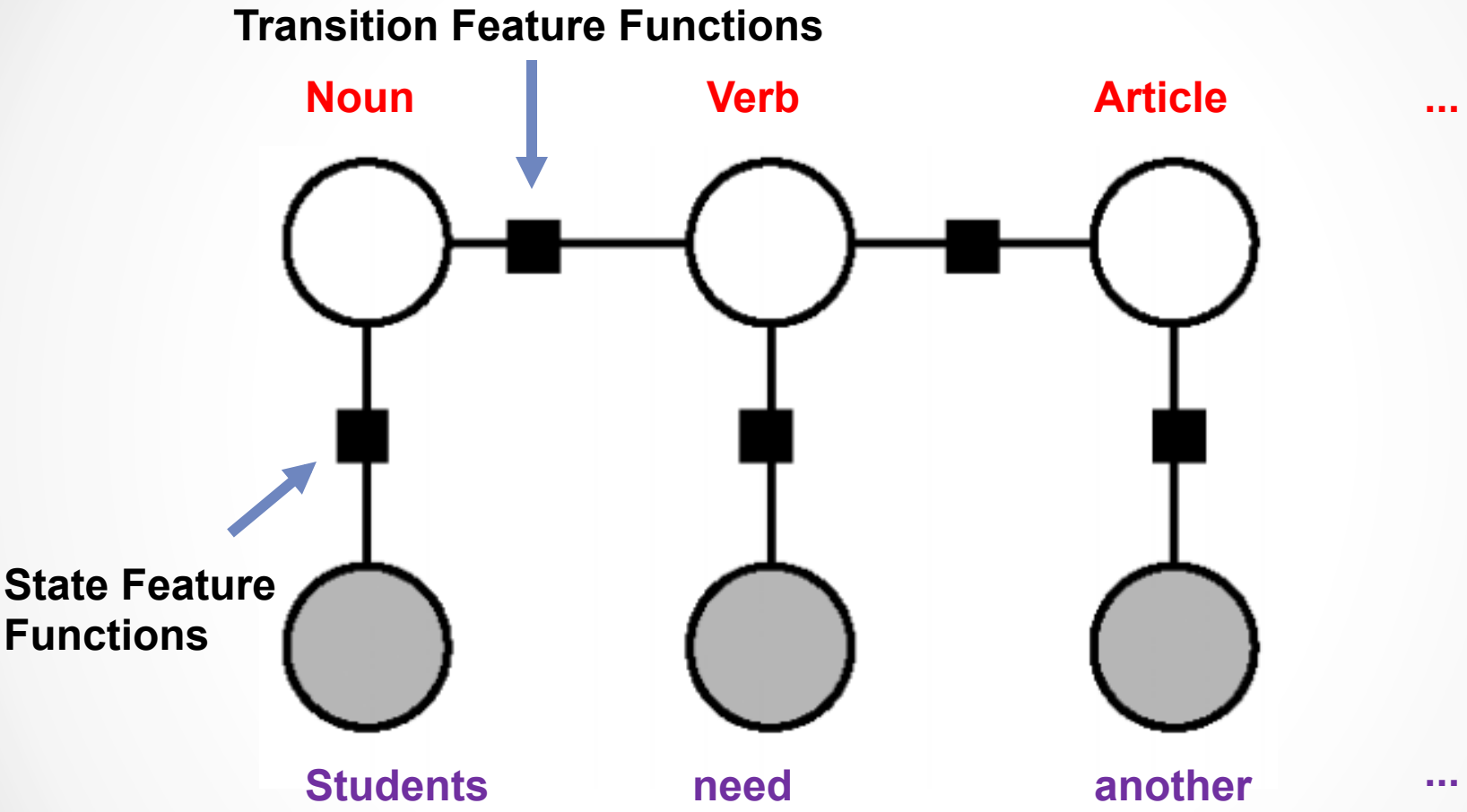
- **POS(part of speech) tagging; the identification of words as nouns, verbs, adjectives, adverbs, etc.**



Using CRF…



Using CRF…


Transition Feature Functions
Noun
Verb
Article
...
State Feature Functions
Students
need
another
…

Some applications: Part-of-Speech-Tagging

- **POS(part of speech) tagging; the identification of words as nouns, verbs, adjectives, adverbs, etc.**

- CRF features:

Feature Type	Description
Transition	$\forall k,k'\ y_i = k$ and $y_{i+1}=k'$

Some applications: Part-of-Speech-Tagging

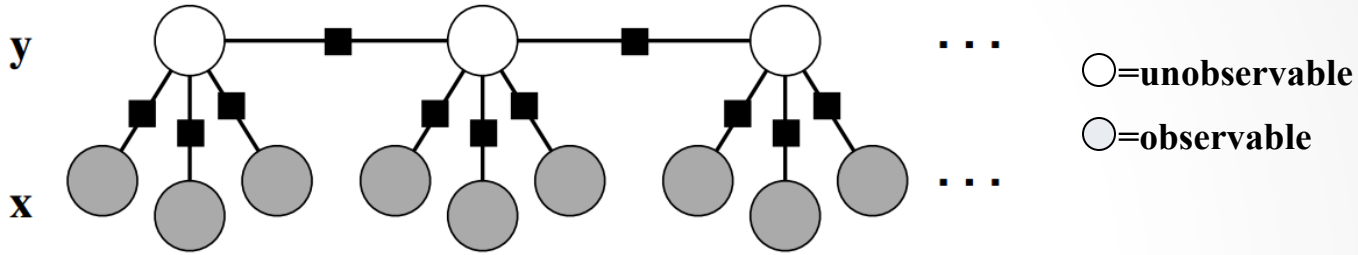
- **POS(part of speech) tagging; the identification of words as nouns, verbs, adjectives, adverbs, etc.**

- CRF features:

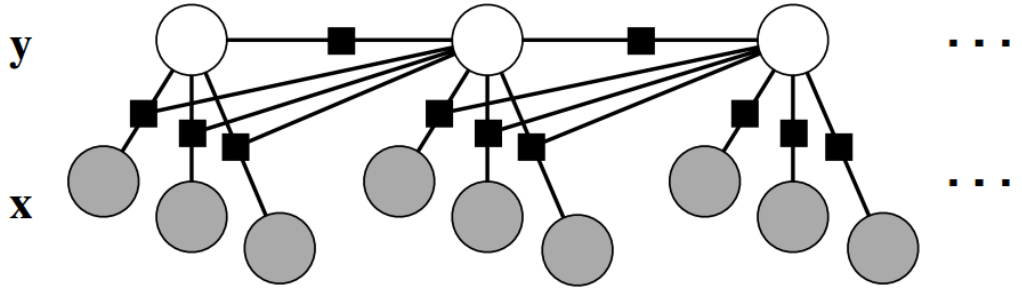
Feature Type	Description
Transition	$\forall k,k'\ y_i = k$ and $y_{i+1}=k'$
State (Word)	$\forall k,w\ y_i = k$ and $x_i=w$ $\forall k,w\ y_i = k$ and $x_{i-1}=w$ $\forall k,w\ y_i = k$ and $x_{i+1}=w$ $\forall k,w,w'\ y_i = k$ and $x_i=w$ and $x_{i-1}=w'$ $\forall k,w,w'\ y_i = k$ and $x_i=w$ and $x_{i+1}=w'$
State (Orthography: Suffix)	$\forall s$ in {"ing","ed","ogy","s","ly","ion","tion", "ity", …} and $\forall k\ y_i=k$ and x_i ends with s
State (Orthography: Punctuation)	$\forall k\ y_i = k$ and x_i is capitalized $\forall k\ y_i = k$ and x_i is hyphenated …

More Complicated (and Expressive) Chain CRFs

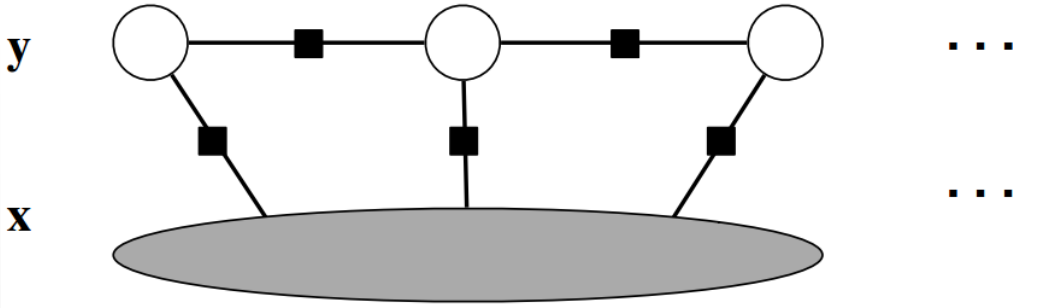
- We can change it so that each state depends on more observations



- Or inputs at previous steps

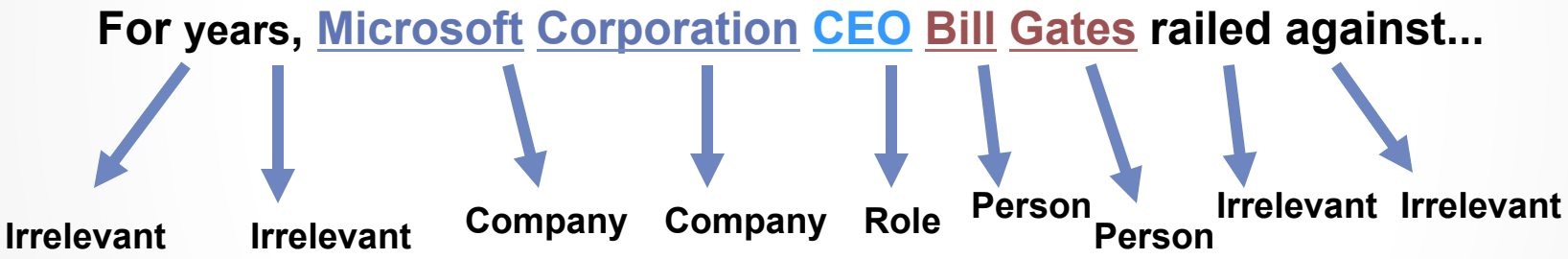


- Or all inputs



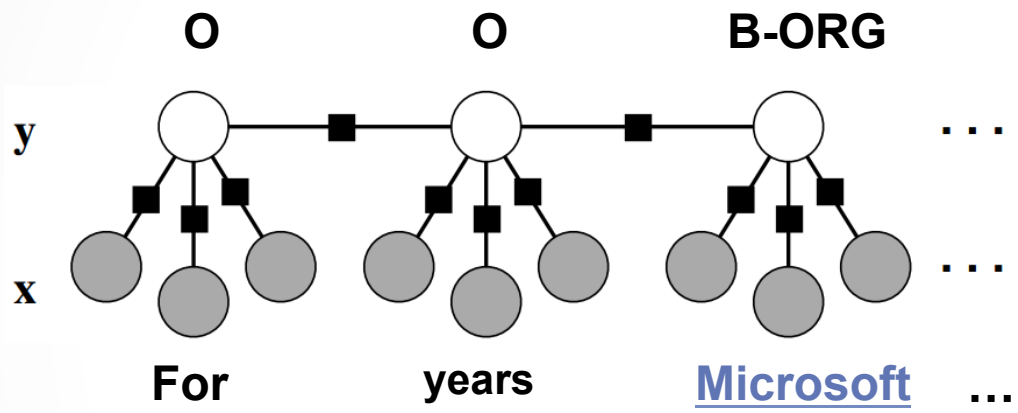
NER as Sequence Labeling Recap

NER can be modeled as sequence labeling!



NER as Sequence Labeling

Features are very important for the CRF model as well. E.g.,



POS tags
Capitalization
Custom-dictionaries
Prefixes
Suffixes
Many more!

…