Social Media & Text Analysis

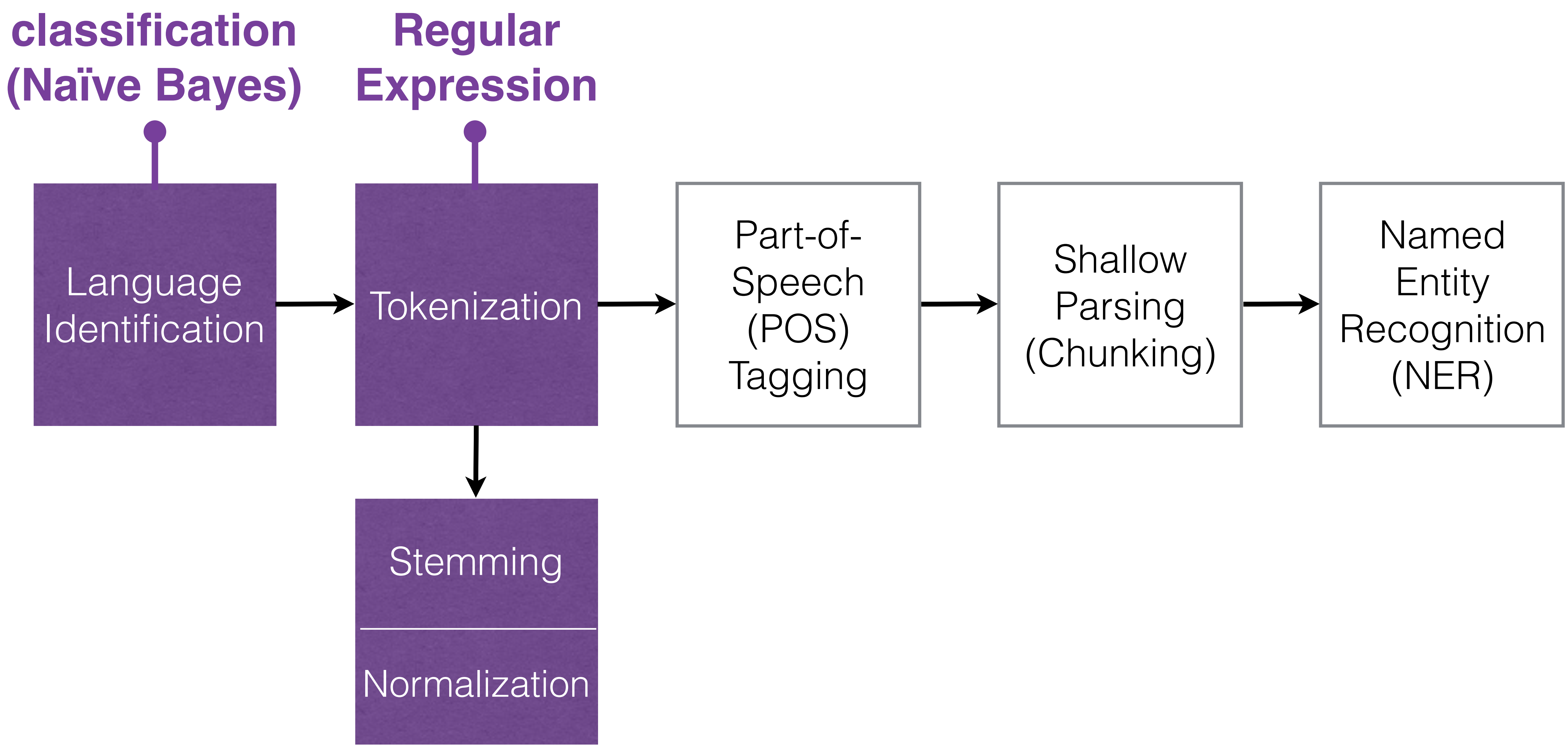
lecture 5 - natural language processing (part 3): POS tagging, chunking, named entity recognition

Follow @cocoweixu

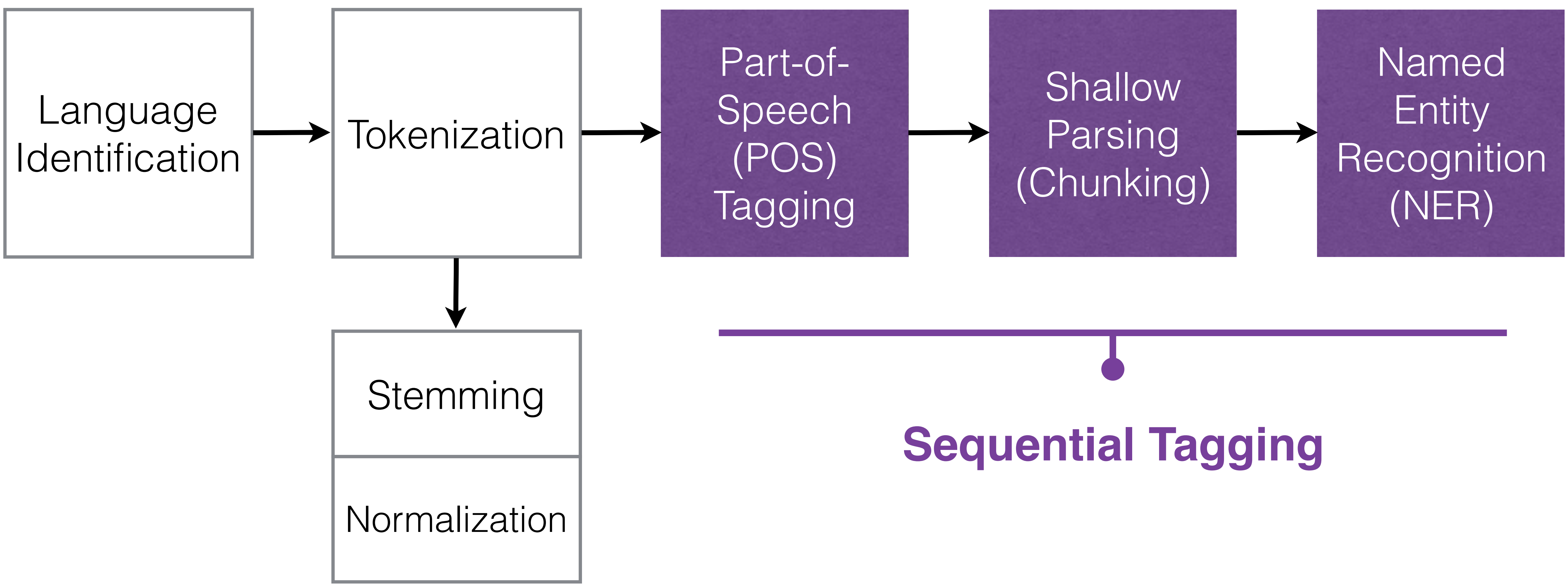
Instructor: Wei Xu

Website: socialmedia-class.org

[Recap] NLP Pipeline



NLP Pipeline



Part-of-Speech (POS) Tagging

Cant	MD
wait	VB
for	IN
the	DT
ravens	NNP
game	NN
tomorrow	NN
...	:
go	VB
ray	NNP
rice	NNP
!!!!!!!	.

Penn Treebank POS Tags

1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition/subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol (mathematical or scientific)
25.	TO	*to*
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund/present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd ps. sing. present
32.	VBZ	Verb, 3rd ps. sing. present
33.	WDT	*wh*-determiner
34.	WP	*wh*-pronoun
35.	WP$	Possessive *wh*-pronoun
36.	WRB	*wh*-adverb
37.	#	Pound sign
38.	$	Dollar sign
39.	.	Sentence-final punctuation
40.	,	Comma
41.	:	Colon, semi-colon
42.	(	Left bracket character
43.	)	Right bracket character
44.	″	Straight double quote
45.	‘	Left open single quote
46.	“	Left open double quote
47.	’	Right close single quote
48.	”	Right close double quote

Part-of-Speech (POS) Tagging

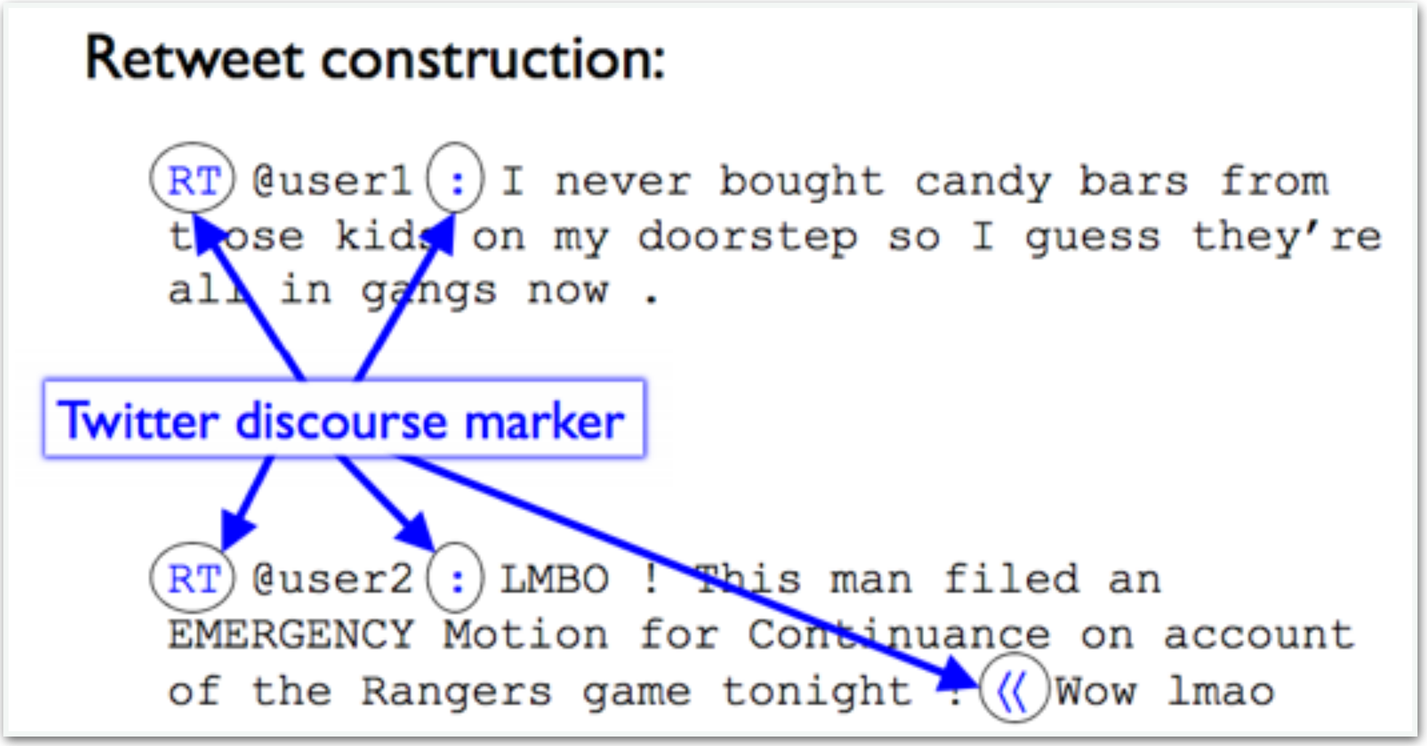
- Words often have more than one POS:
  - The <u>back</u> door = JJ
  - On my <u>back</u> = NN
  - Win the voters <u>back</u> = RB
  - Promised to <u>back</u> the bill = VB

- POS tagging problem is to determine the POS tag for a particular instance of a word.

Source: adapted from Chris Manning

Twitter-specific Tags

- #hashtag
- @metion
- url
- email address
- emoticon
- discourse marker
- symbols
- …



Source: Gimpel et al.
"Part-of-Speech Tagging for Twitter : Annotation, Features, and Experiments" ACL 2011

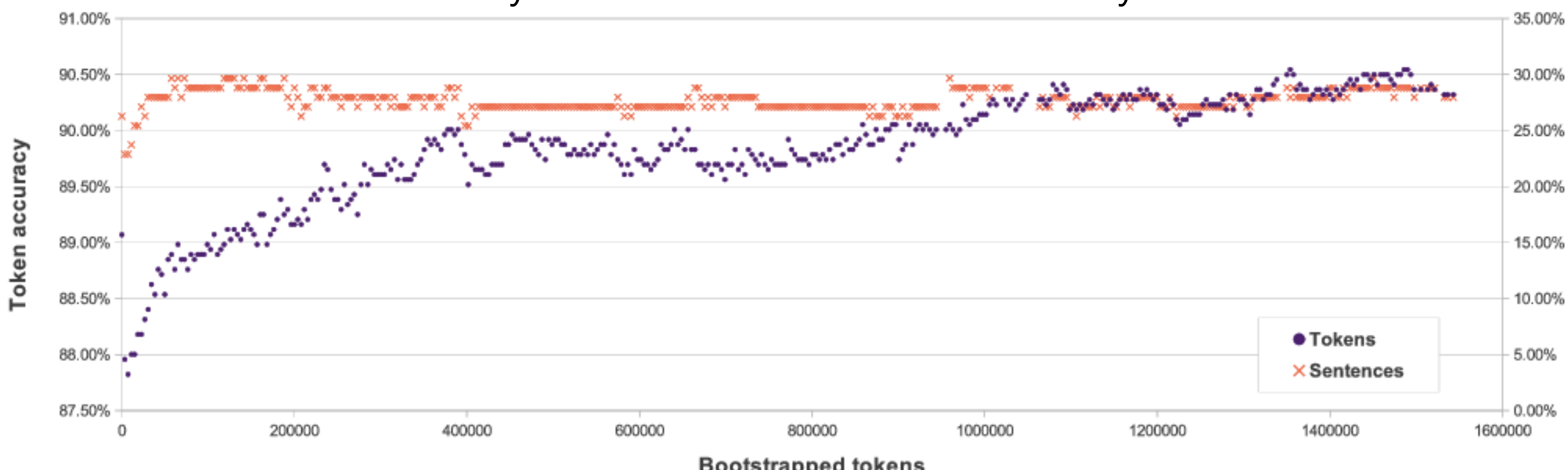
Notable Twitter POS Taggers

- Gimpel et al., 2011
- Ritter et al., 2011
- Derczynski et al, 2013
- Owoputi et al. 2013

(97% on news text)

State-of-the-art:

Token Accuracy: ~ 88% Sentence Accuracy ~20%



Source: Derczynski, Ritter, Clark, Bontcheva
“Twitter Part-of-Speech Tagging for All: Overcoming Sparse and Noisy Data" RANLP 2013

Chunking

Cant	VP
wait	
for	PP
the	NP
ravens	
game	
tomorrow	NP
…	
go	VP
ray	NP
rice	
!!!!!!!	

Chunking

- recovering phrases constructed by the part-of-speech tags
- a.k.a shallow (partial) parsing:
    - full parsing is expensive, and is not very robust
    - partial parsing can be much faster, more robust, yet sufficient for many applications
    - useful as input (features) for named entity recognition or full parser

Named Entity Recognition(NER)

Cant	
wait	
for	
the	
ravens	ORG
game	
tomorrow	
…	
go	
ray	PER
rice	PER
!!!!!!!	.

ORG: organization

PER: person

LOC: location

NER: Basic Classes

Cant	
wait	
for	
the	
ravens	ORG
game	
tomorrow	
...	
go	
ray	PER
rice	
!!!!!!!	.

ORG: organization

PER: person

LOC: location

NER: Rich Classes

Source: Baldwin, de Marneffe, Han, Kim, Ritter, Xu
Shared Tasks of the 2015 Workshop on Noisy User-generated Text:
Twitter Lexical Normalization and Named Entity Recognition

NER: Genre Differences

	News	Tweets
PER	Politicians, business leaders, journalists, celebrities	Sportsmen, actors, TV personalities, celebrities, names of friends
LOC	Countries, cities, rivers, and other places related to current affairs	Restaurants, bars, local landmarks/areas, cities, rarely countries
ORG	Public and private companies, government organisations	Bands, internet companies, sports clubs

Source: Kalina Bontcheva and Leon Derczynski
"Tutorial on Natural Language Processing for Social Media" EACL 2014

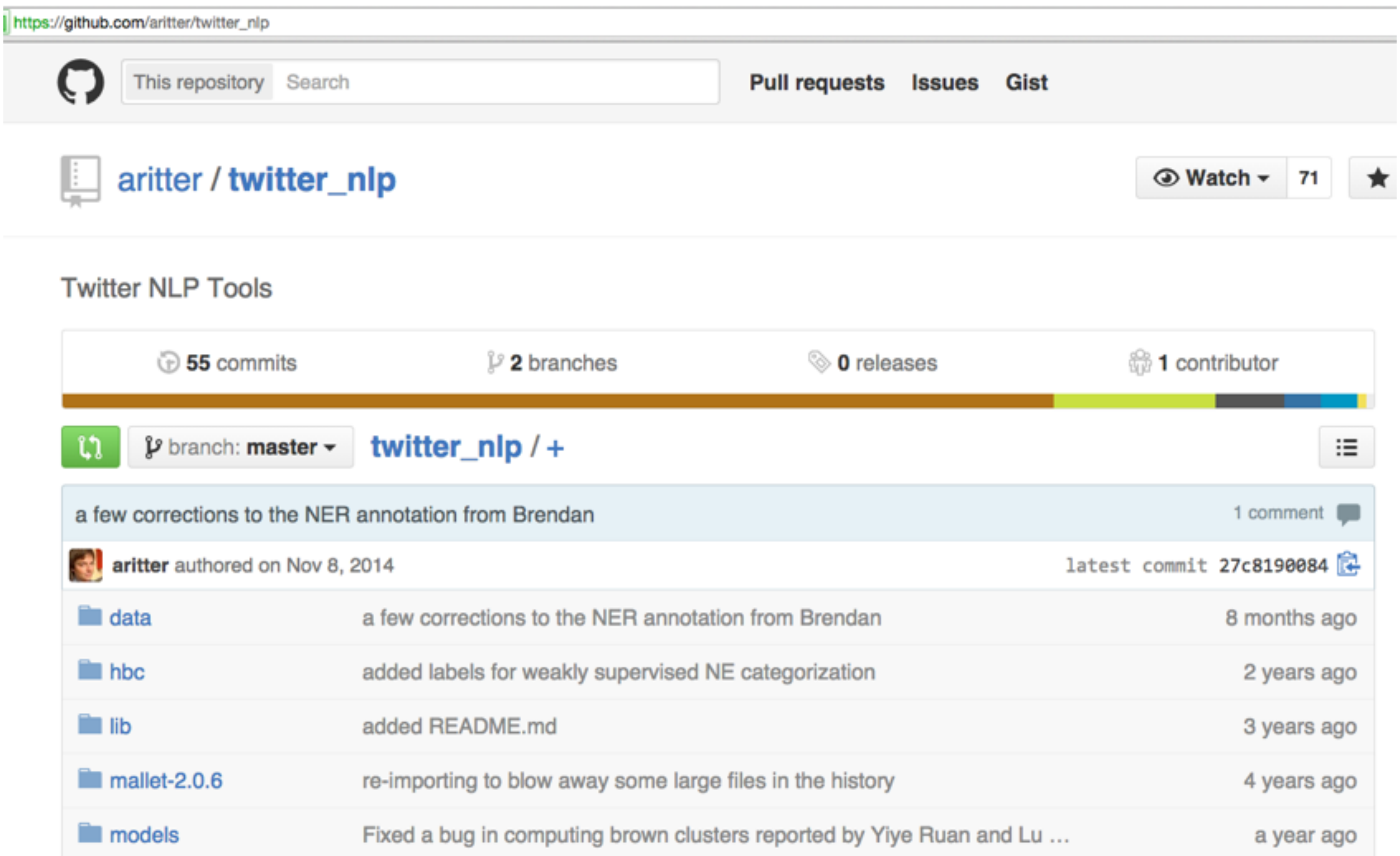
Notable Twitter NE Research

- Liu et al., 2011
- Ritter et al., 2011
- Owoputi et al. 2013
- Plank et al, 2014
- Cherry & Guo, 2015

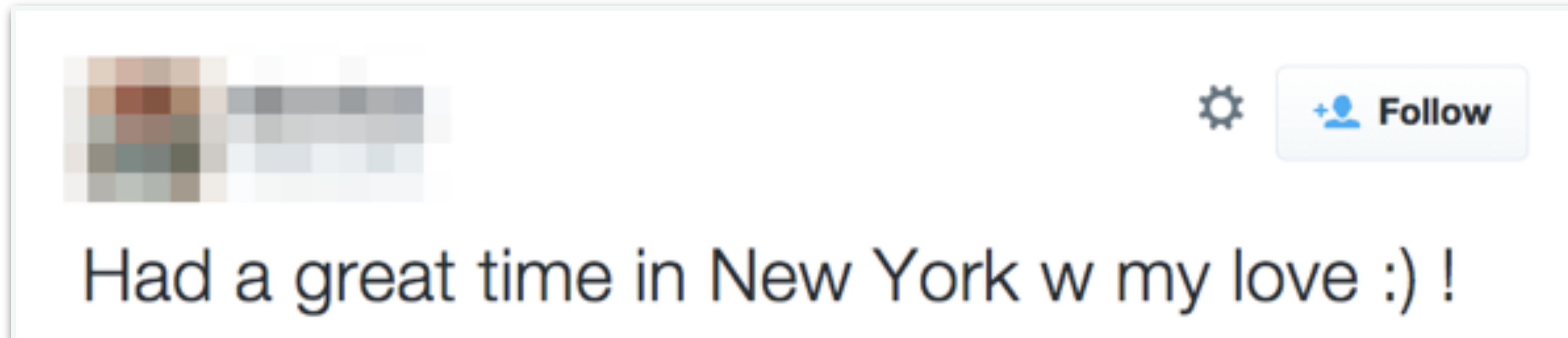
System	P	R	F_1
COTRAIN-NER (10 types)	0.55	0.33	0.41
T-NER(10 types)	0.65	0.42	**0.51**
COTRAIN-NER (PLO)	0.57	0.42	0.49
T-NER(PLO)	0.73	0.49	**0.59**
Stanford NER (PLO)	0.30	0.27	0.29

Table 12: Performance at predicting both segmentation and classification. Systems labeled with PLO are evaluated on the 3 MUC types *PERSON*, *LOCATION*, *ORGANIZATION*.

Tool: twitter_nlp



Tool: twitter_nlp




```
xuwei@proteus100[twitter_nlp]$ export TWITTER_NLP=./
xuwei@proteus100[twitter_nlp]$
xuwei@proteus100[twitter_nlp]$ echo "Had a great time in New York w my
love :) ! " | python python/ner/extractEntities2.py

Had/O a/O great/O time/O in/O New/B-ENTITY York/I-ENTITY w/O my/O love/
O :)/O !/O
Average time per tweet = 3.04769945145s
xuwei@proteus100[twitter_nlp]$
xuwei@proteus100[twitter_nlp]$ echo "Had a great time in New York w my
love :) ! " | python python/ner/extractEntities2.py --pos --chunk

Had/O/VBD/B-VP a/O/DT/B-NP great/O/JJ/I-NP time/O/NN/I-NP in/O/IN/B-PP
New/B-ENTITY/NNP/B-NP York/I-ENTITY/NNP/I-NP w/O/IN/B-PP my/O/PRP$/B-NP
 love/O/NN/I-NP :)/O/UH/B-INTJ !/O/./I-INTJ
Average time per tweet = 5.49846148491s
xuwei@proteus100[twitter_nlp]$ _
```

IO tag encoding

Cant	VP		
wait			
for	PP		
the	NP		
ravens			
game			
tomorrow	NP		
…			
go	VP		
ray	NP		
rice			
!!!!!!!			

IO tag encoding

Cant	VP	VP	
wait		VP	
for	PP	PP	
the		NP	
ravens	NP	NP	
game		NP	
tomorrow	NP	NP	
…		O	
go	VP	VP	
ray	NP	NP	
rice		NP	
!!!!!!!		O	

IO tag encoding

Cant	VP	VP	B-VP
wait		VP	I-VP
for	PP	PP	B-PP
the	NP	NP	B-NP
ravens		NP	I-NP
game		NP	I-NP
tomorrow	NP	NP	B-NP
…		O	O
go	VP	VP	B-VP
ray	NP	NP	B-VP
rice		NP	I-VP
!!!!!!!		O	O

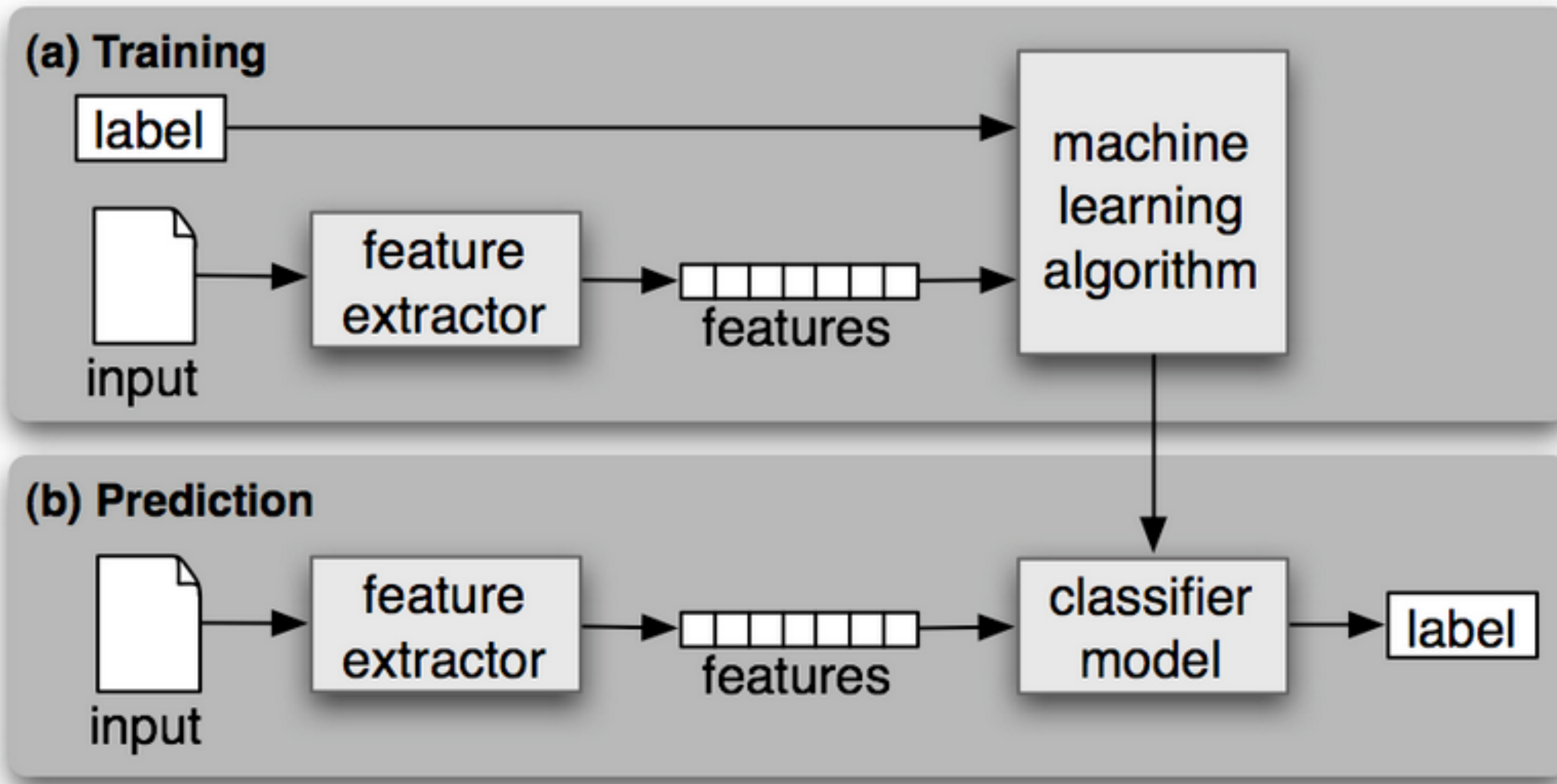
I: Inside

O: outside

B: Begin

BIO allows separation of adjacent chunks/entities

[Recap] Classification Method:
Supervised Machine Learning



Source: NLTK Book

[Recap] Classification Method:

Supervised Machine Learning

- Input:
  - a document d
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_j\}$
  - a training set of m hand-labeled documents $(d_1, c_1), \ldots, (d_m, c_m)$
- Output:
  - a learned classifier $\gamma : d \rightarrow c$

[Recap] Classification Method:
Supervised Machine Learning

- Naïve Bayes
- Logistic Regression
- Support Vector Machines (SVM)
- …

[Recap] Naïve Bayes

- **Conditional Independence Assumption**:

features ***P(t|c)*** are independent given the class ***c***

$$P(t_1, t_2, \ldots, t_n \mid c)$$
$$= P(t_1 \mid c) \cdot P(t_2 \mid c) \cdot \ldots \cdot P(t_n \mid c)$$

Source: adapted from Dan jurafsky

[Recap] Bag-of-Words

- **positional independence assumption**:
  - features are the words occurring in the document and their value is the number of occurrences
  - word probabilities are position independent

Classification Method:

Supervised Machine Learning

- Naïve Bayes
- Logistic Regression
- Support Vector Machines (SVM)
- …
- Hidden Markov Model (HMM)
- Conditional Random Fields (CRF)
- …

sequential models

Classification Method:

Sequential Supervised Learning

- Input:
  - rather than just individual examples ***(w_1 = the, c_1 = DT)***
  - a training set consists of ***m*** sequences of labeled examples ***(x_1, y_1), … , (x_m, y_m)***

    x_1 = <the back door> and y_1= <DT JJ NN>

- Output:
  - a learned classifier to predict label sequences ***$\gamma : x \rightarrow y$***

Features for Sequential Tagging

- Words:
  - current words
  - previous/next word(s) — context
- Other linguistic information:
  - word substrings
  - word shapes
  - POS tags
- Contextual Labels
  - previous (and perhaps next) labels

word shapes

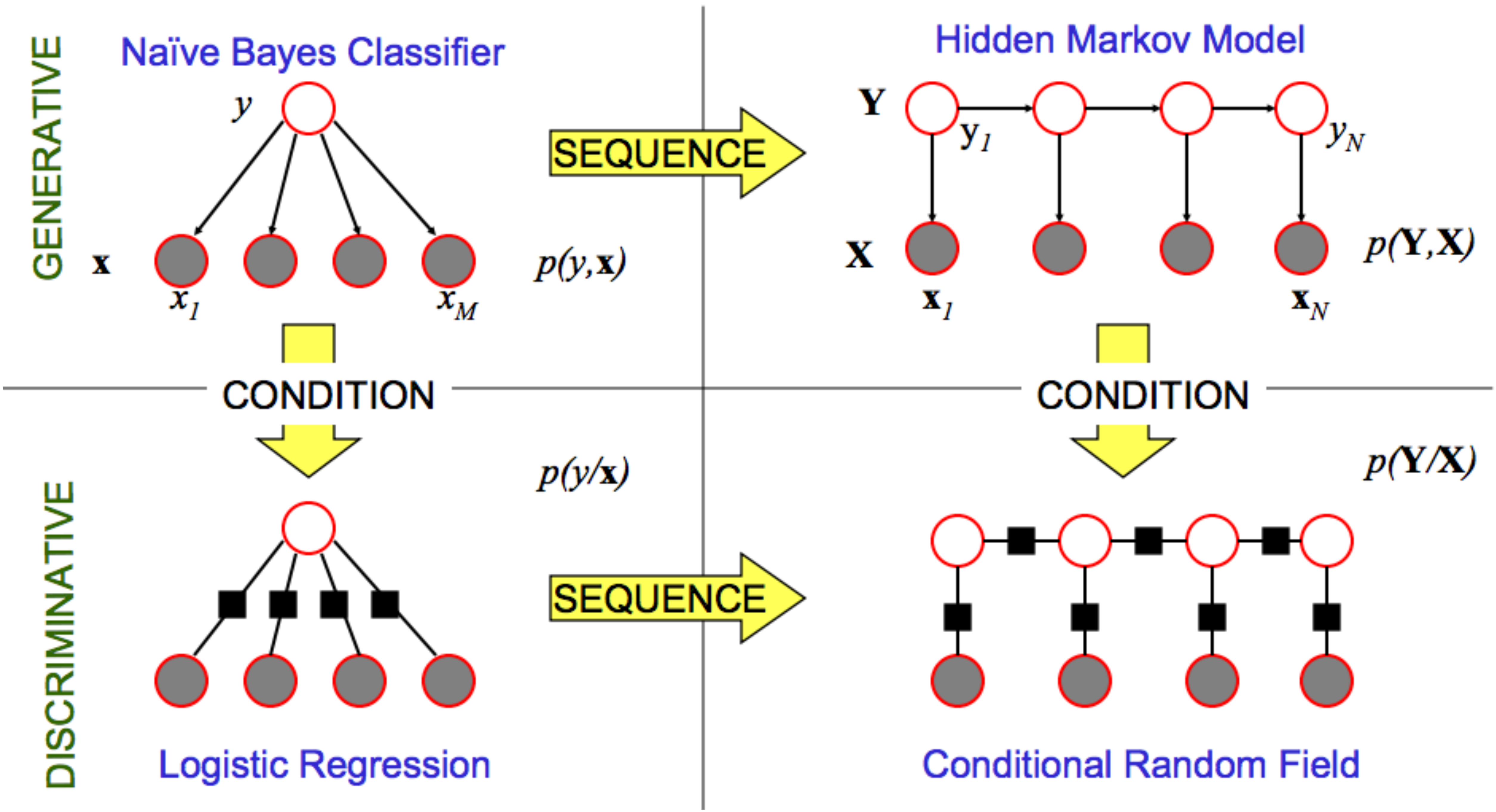
Varicella-zoster	Xx-xxx
mRNA	xXXX
CPA1	XXXd

Features for Sequential Tagging

- Words:
  - current words
  - previous/next word(s) — context
- Other linguistic information:
  - word substrings
  - word shapes
  - POS tags
- Contextual Labels
  - previous (and perhaps next) labels

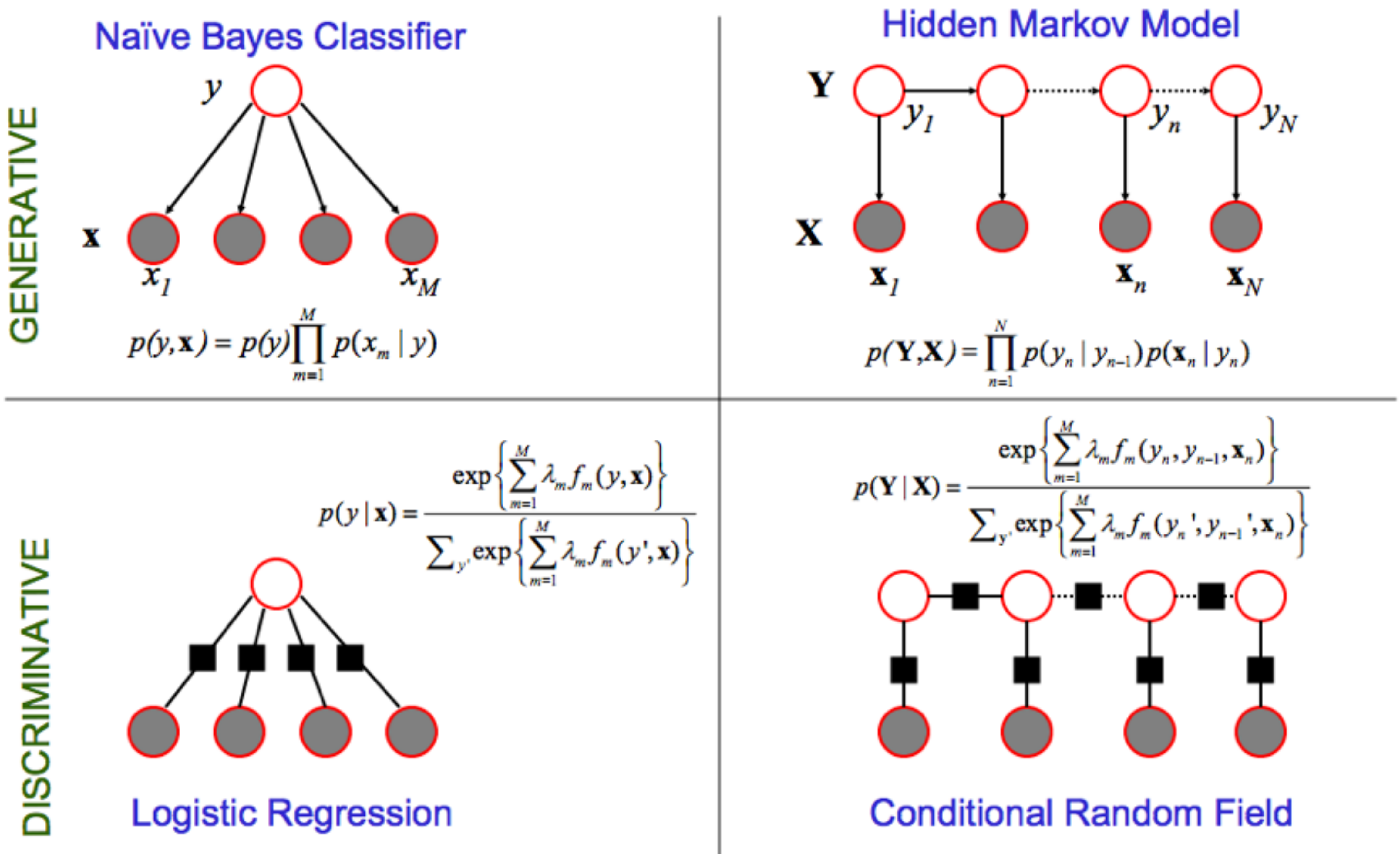
Correlated!

Graphical Models



Source: Andrew McCallum, Sargur Srihari

Graphical Models



Source: Andrew McCallum, Sargur Srihari

Twitter Challenge

2m 2ma 2mar 2mara 2maro 2marrow 2mor 2mora 2moro 2morow 2morr 2morro 2morrow 2moz 2mr 2mro 2mrrw 2mrw 2mw tmmrw tmo tmoro tmorrow tmoz tmr tmro tmrow tmrrow tmrrw tmrw tmrww tmw tomaro tomarow tomarro tomarrow tomm tommarow tommarrow tommoro tommorow tommorrow tommorw tommrow tomo tomolo tomoro tomorow tomorro tomorrw tomoz tomrw tomz

An Unsupervised Learning Method:

Brown Clustering

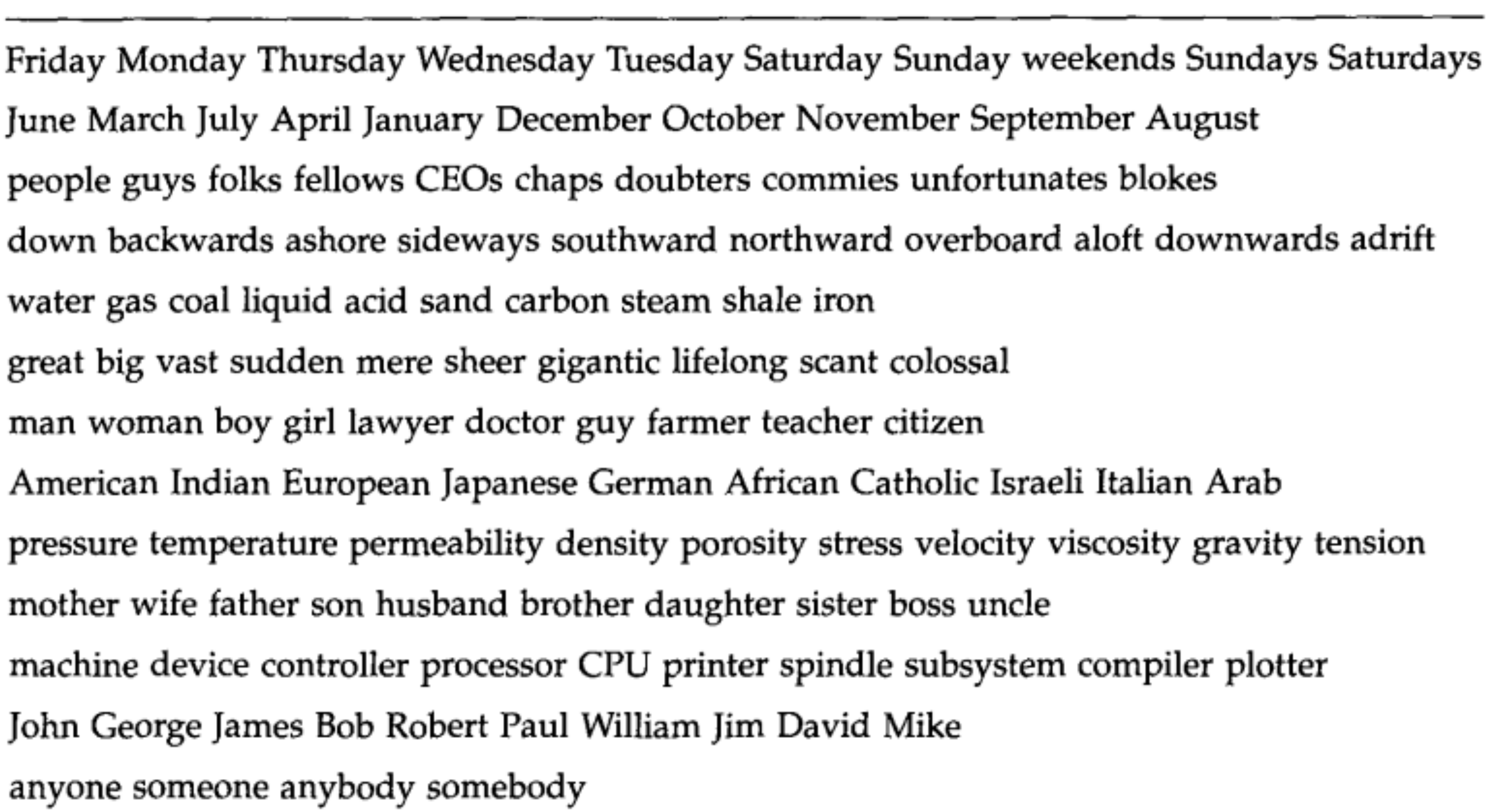
- Input:
  - a (large) corpus of documents

- Output:

  1. a partition of words into word clusters
  2. (generalization of 1) a hierarchical word clustering

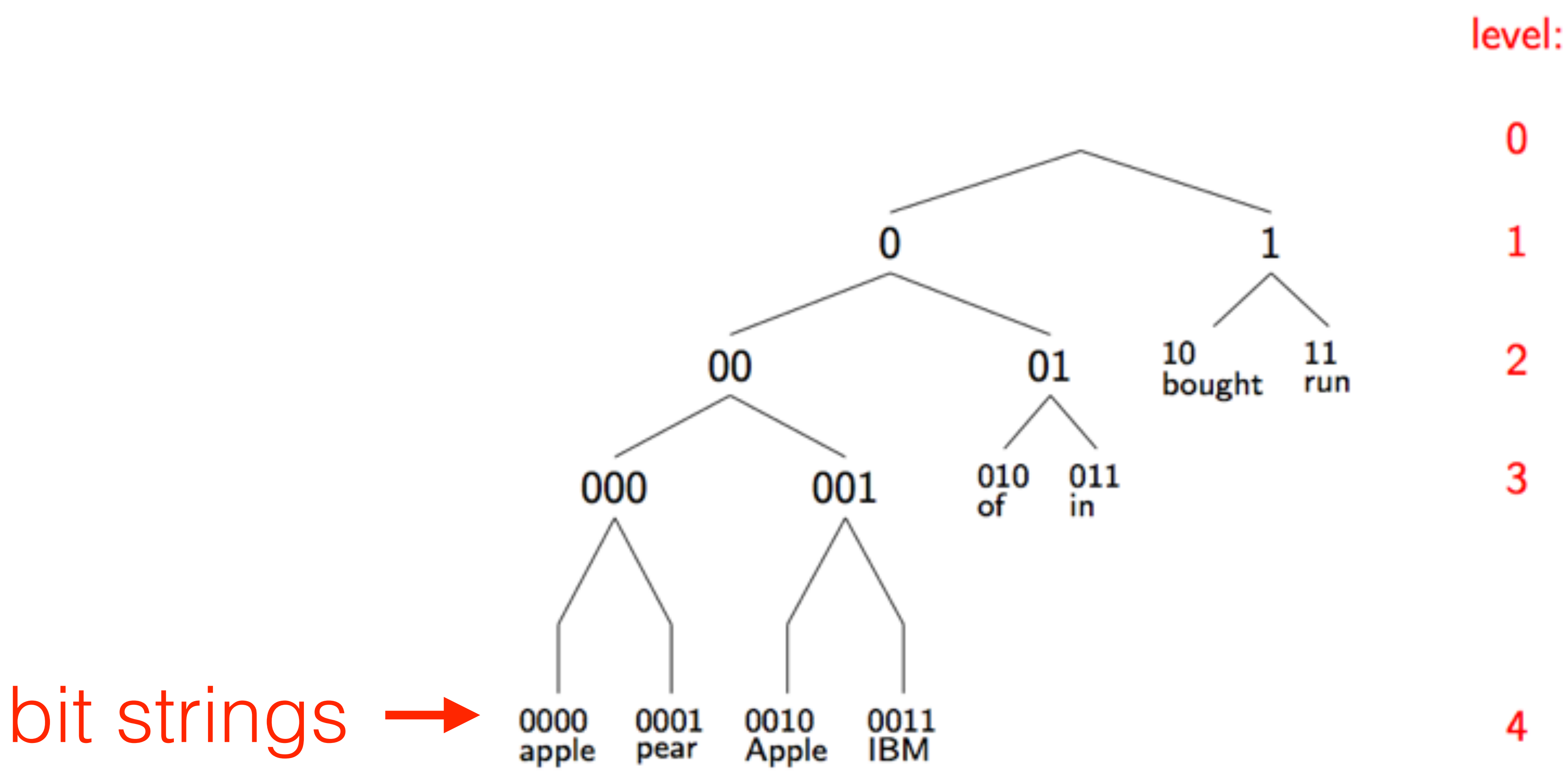
Brown Clustering

- Example Clusters (from Brown et al. 1992)


Friday Monday Thursday Wednesday Tuesday Saturday Sunday weekends Sundays Saturdays
June March July April January December October November September August
people guys folks fellows CEOs chaps doubters commies unfortunates blokes
down backwards ashore sideways southward northward overboard aloft downwards adrift
water gas coal liquid acid sand carbon steam shale iron
great big vast sudden mere sheer gigantic lifelong scant colossal
man woman boy girl lawyer doctor guy farmer teacher citizen
American Indian European Japanese German African Catholic Israeli Italian Arab
pressure temperature permeability density porosity stress velocity viscosity gravity tension
mother wife father son husband brother daughter sister boss uncle
machine device controller processor CPU printer spindle subsystem compiler plotter
John George James Bob Robert Paul William Jim David Mike
anyone someone anybody somebody

Hierarchical Word Clustering

- bit string representation:



Hierarchical Word Clustering

mailman	10000011010111
salesman	100000110110000
bookkeeper	1000001101100010
troubleshooter	10000011011000110
bouncer	10000011011000111
technician	1000001101100100
janitor	1000001101100101
saleswoman	1000001101100110
...	
Nike	1011011100100101011100
Maytag	10110111001001010111010
Generali	10110111001001010111011
Gap	1011011100100101011110
Harley-Davidson	10110111001001010111110
Enfield	101101110010010101111110
genus	101101110010010101111111
Microsoft	10110111001001011000
Ventritex	101101110010010110010
Tractebel	1011011100100101100110
Synopsys	1011011100100101100111
WordPerfect	1011011100100101101000
....	
John	101110010000000000
Consuelo	101110010000000001
Jeffrey	101110010000000010
Kenneth	10111001000000001100
Phillip	101110010000000011010
WILLIAM	101110010000000011011
Timothy	10111001000000001110

- Example Clusters (from Miller et al. 2014)

Hierarchical Word Clustering

mailman	10000011010111
salesman	100000110110000
bookkeeper	1000001101100010
troubleshooter	10000011011000110
bouncer	10000011011000111
technician	1000001101100100
janitor	1000001101100101
saleswoman	1000001101100110
...	
Nike	1011011100100101011100
Maytag	10110111001001010111010
Generali	10110111001001010111011
Gap	1011011100100101011110
Harley-Davidson	10110111001001010111110
Enfield	101101110010010101111110
genus	101101110010010101111111
Microsoft	10110111001001011000
Ventritex	101101110010010110010
Tractebel	1011011100100101100110
Synopsys	1011011100100101100111
WordPerfect	1011011100100101101000
....	
John	101110010000000000
Consuelo	101110010000000001
Jeffrey	101110010000000010
Kenneth	1011100100000001100
Phillip	10111001000000011010
WILLIAM	10111001000000011011
Timothy	1011100100000001110

- Example Clusters (from Miller et al. 2014)

word cluster features
(bit string prefix)

Brown Clustering

- The Intuition:
  - similar words appear in similar contexts
  - more precisely:

    similar words have similar distributions of words to their immediate left and right

Brown Clustering

- The algorithm — maximize the Quality function that score a given partitioning ***C*** :

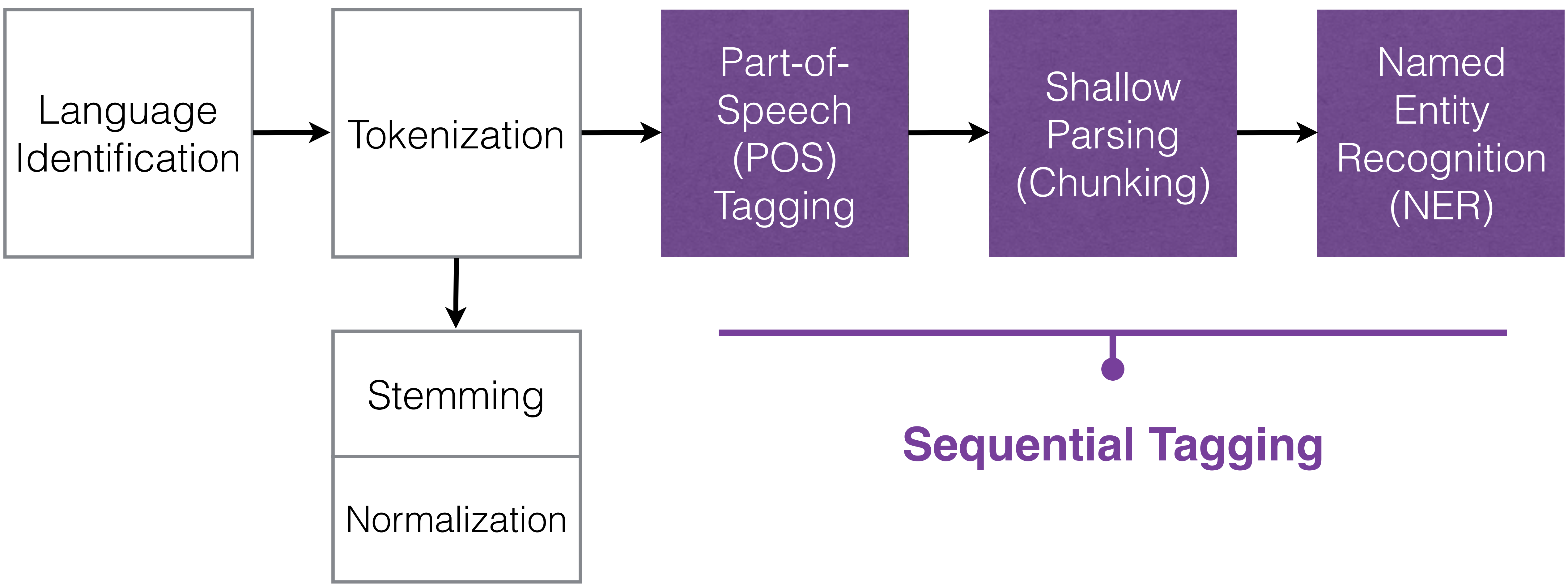
$$Quality(C) = \sum_{i}^{n} \log e(w_i \mid C(w_i)) q(C(w_i) \mid C(w_{i-1}))$$

$$= \sum_{c=1}^{k} \sum_{c'=1}^{k} p(c,c') \log \frac{p(c,c')}{p(c)p(c')} + G$$

  - ***n(c)*** :count of class ***c*** seen in the corpus
  - ***n(c,c')*** : counts of ***c'*** seen following ***c***

$$p(c,c') = \frac{n(c,c')}{\sum_{c,c'} n(c,c')} \qquad p(c,c') = \frac{n(c)}{\sum_{c} n(c)}$$

Summary


Language Identification
Tokenization
Part-of-Speech (POS) Tagging
Shallow Parsing (Chunking)
Named Entity Recognition (NER)
Stemming
Normalization
Sequential Tagging

Thank You!

Follow @cocoweixu

Instructor: Wei Xu

www.cis.upenn.edu/~xwe/

Course Website: socialmedia-class.org