Sequence Labeling II

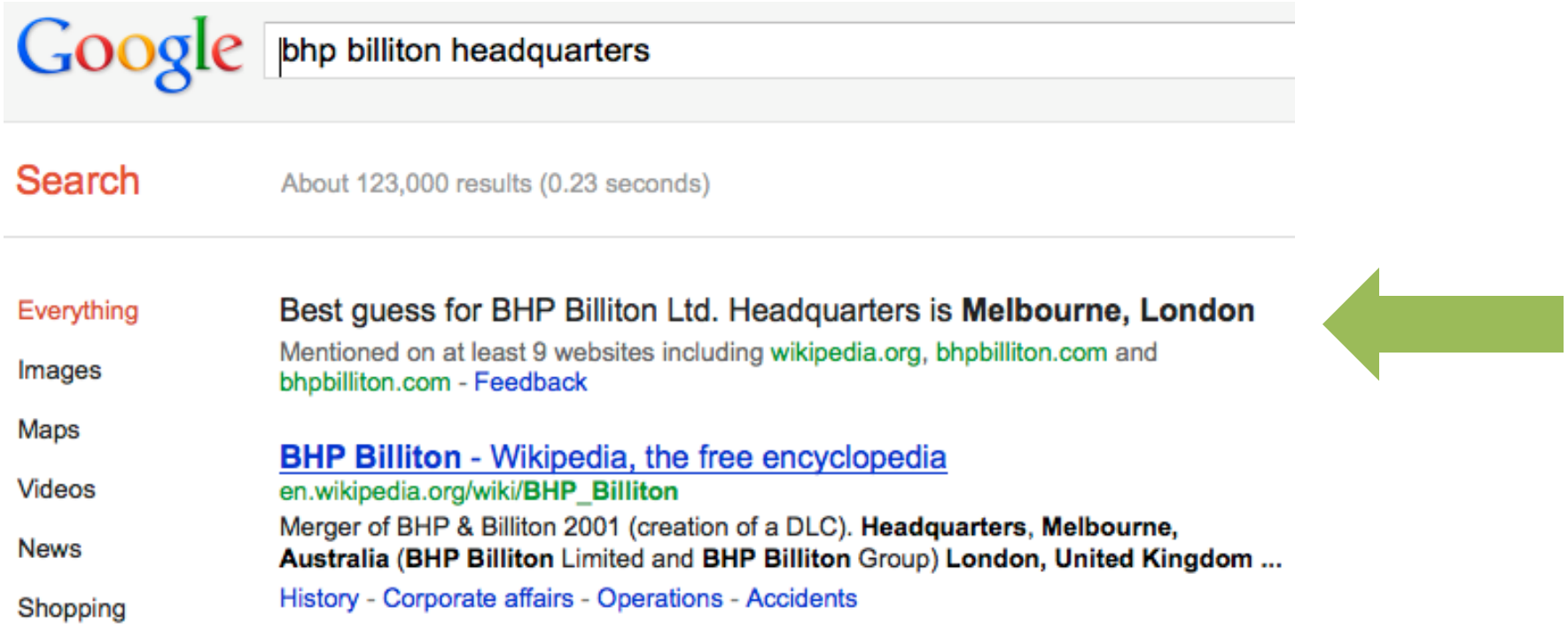
Named Entity Recognition with Conditional Random Fields

Mausam

(Slides based on Ray Mooney, Ramesh Nallapati, Alan Ritter, Fei Xia, Alex Yates, Michael Collins, Heng Ji, Dan Jurafsky, Dan Klein, Chris Manning, Lev Ratinov, Luke Zettlemoyer)

Named Entity Recognition

Information Extraction



Named Entity Recognition (NER)

- A very important sub-task: find and classify names in text, for example:
  - The decision by the independent MP Andrew Wilkie to withdraw his support for the minority Labor government sounded dramatic but it should not further threaten its stability. When, after the 2010 election, Wilkie, Rob Oakeshott, Tony Windsor and the Greens agreed to support Labor, they gave just two guarantees: confidence and supply.

Named Entity Recognition (NER)

- A very important sub-task: find and classify names in text, for example:
  - The decision by the independent MP Andrew Wilkie to withdraw his support for the minority Labor government sounded dramatic but it should not further threaten its stability. When, after the 2010 election, Wilkie, Rob Oakeshott, Tony Windsor and the Greens agreed to support Labor, they gave just two guarantees: confidence and supply.

Named Entity Recognition (NER)

Person
Date
Location
Organi-
zation

- A very important sub-task: find and classify names in text, for example:
  - The decision by the independent MP Andrew Wilkie to withdraw his support for the minority Labor government sounded dramatic but it should not further threaten its stability. When, after the 2010 election, Wilkie, Rob Oakeshott, Tony Windsor and the Greens agreed to support Labor, they gave just two guarantees: confidence and supply.

Named Entity Recognition (NER)

- The uses:
  - Named entities can be indexed, linked off, etc.
  - Sentiment can be attributed to companies or products
  - A lot of IE relations are associations between named entities
  - For question answering, answers are often named entities.
- Concretely:
  - Many web pages tag various entities, with links to bio or topic pages, etc.
    - Reuters' OpenCalais, Evri, AlchemyAPI, Yahoo's Term Extraction, …
  - Apple/Google/Microsoft/… smart recognizers for document content

The Named Entity Recognition Task

Task: Predict entities in a text

Foreign	ORG
Ministry	ORG
spokesman	O
Shen	PER
Guofang	PER
told	O
Reuters	ORG
:	:

} Standard evaluation is per entity, *not* per token

Precision/Recall/F1 for IE/NER

- Recall and precision are straightforward for tasks like IR and text categorization, where there is only one grain size (documents)
- The measure behaves a bit funnily for IE/NER when there are *boundary errors* (which are *common*):
  - First Bank of Chicago announced earnings …
- This counts as both a fp and a fn
- Selecting *nothing* would have been better
- Some other metrics (e.g., MUC scorer) give partial credit (according to complex rules)

Sequence model approach to NER

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 sequence classifier to predict the labels from the data

Testing

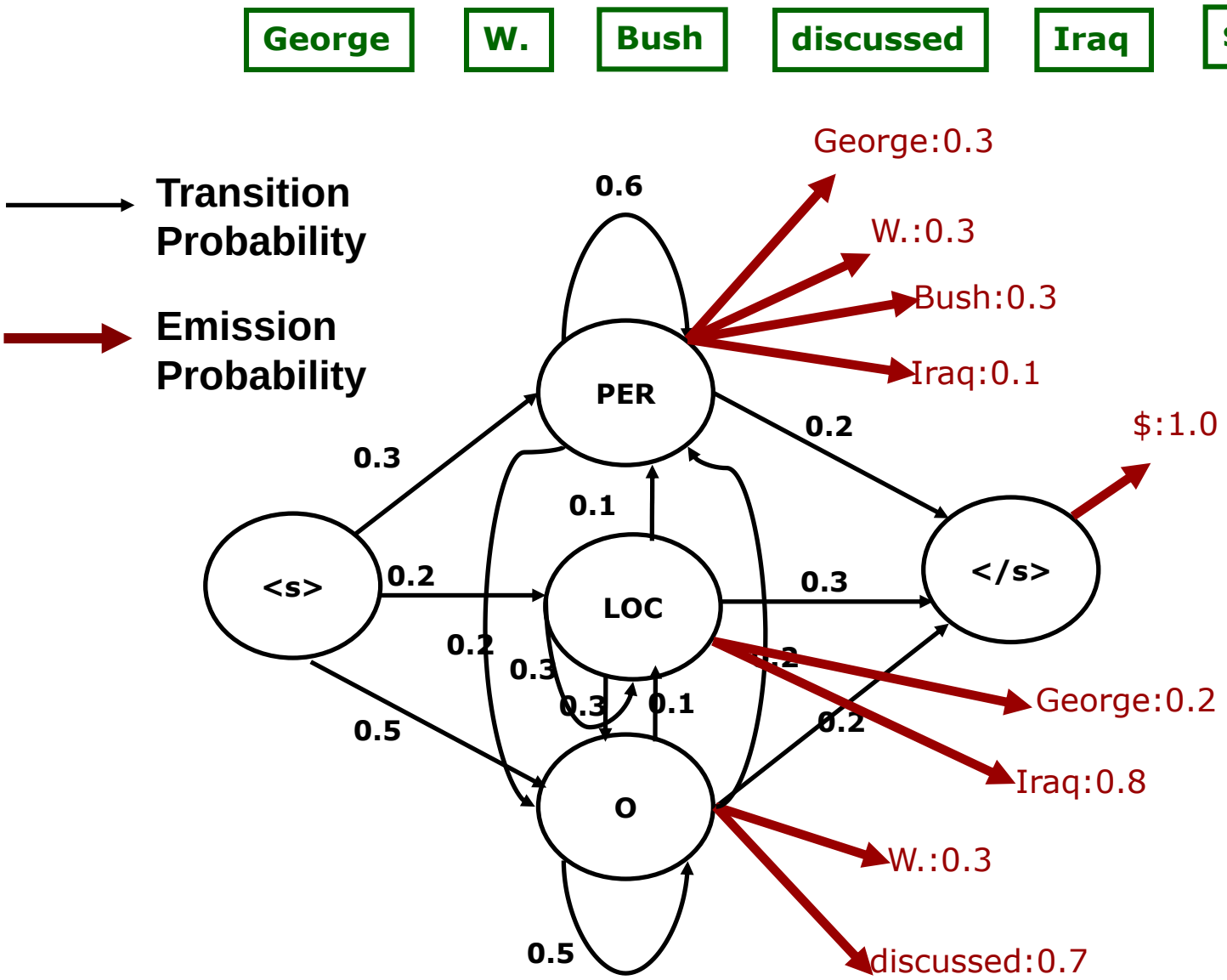
1. Receive a set of testing documents
2. Run sequence model inference to label each token
3. Appropriately output the recognized entities

Encoding classes for NER

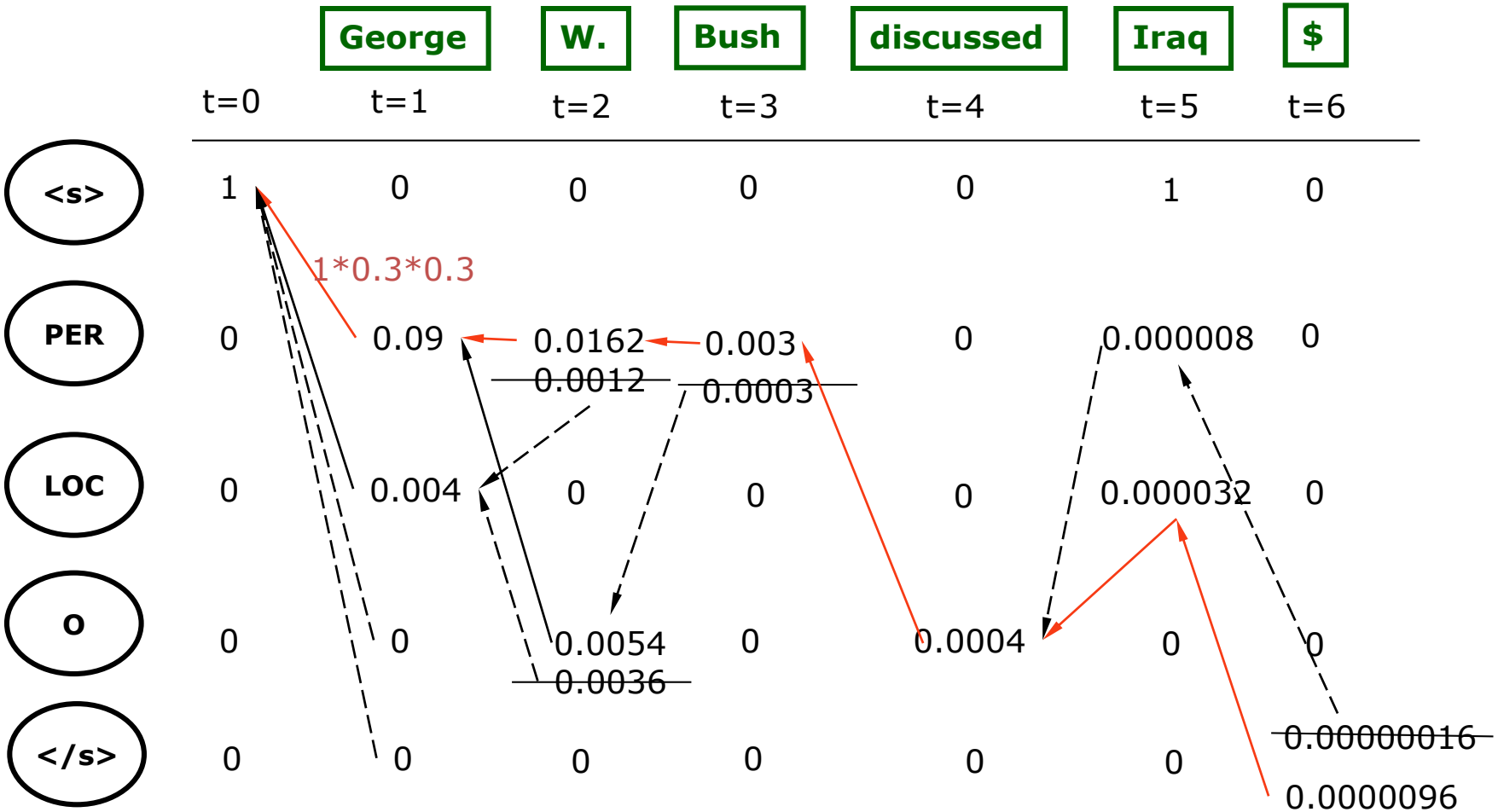
	IO encoding	IOB encoding
Fred	PER	B-PER
showed	O	O
Sue	PER	B-PER
Mengqiu	PER	B-PER
Huang	PER	I-PER
‘s	O	O
new	O	O
painting	O	O

Practically negligible differences in performance. IO much faster.

Markov Chain for a Simple Name Tagger



Viterbi Decoding of Name Tagger



Current = Previous * Transition * Emission

Limitations of HMMs

- Modeling more than necessary
  - joint probability distribution $p(y, x)$
- Assumes independent features
- Cannot represent overlapping features or long range dependences between observed elements
  - Need to enumerate all possible observation sequences
  - Very strict independence assumptions on the observations

- Toward discriminative/conditional models
  - Conditional probability P(label sequence y | observation sequence x) rather than joint probability P(y, x)
  - Allow arbitrary, non-independent features on the observation sequence X
  - The probability of a transition between labels may depend on past and future observations
  - Relax strong independence assumptions in generative models

Features for Sequence Labeling

Representation: History

- US/L president/O Obama/P visited/O Delhi/L to/O meet/O with/O Narendra/P Modi/P.
- Define: History -- a 3 tuple < y_{-1}, $x_{[1..T]}$, i>
  - y_{-1}: previous tag
  - $x_{[1..T]}$: all words in the sentence
  - i: index of the word being tagged
- Example: History(Obama)
  - <O,US…Modi,3>

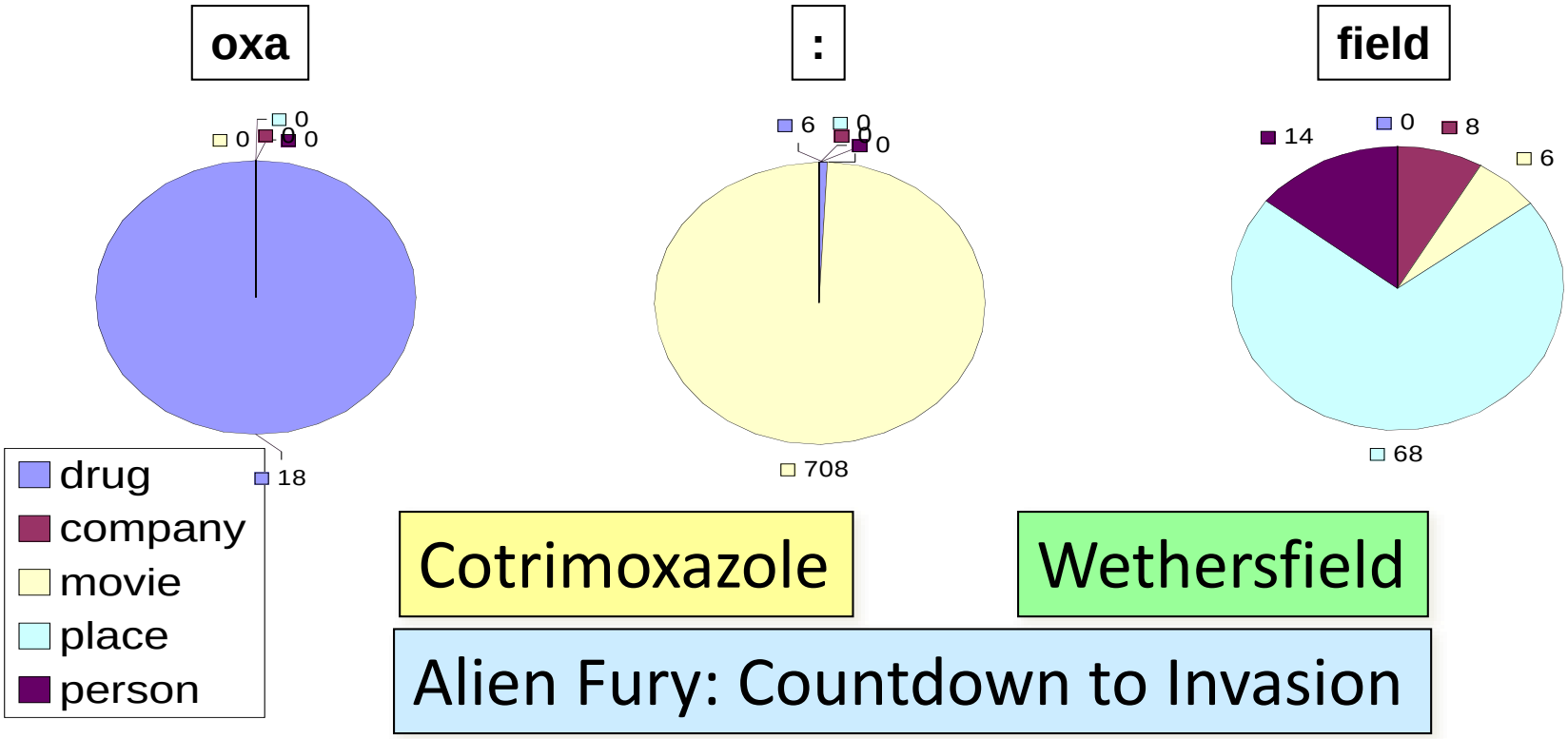
Features for CRFs

- Goal: to define $P(Y|X)$ using features
- Feature is a function ϕ: $H \times Y \rightarrow R$
  - often indicators ($H \times Y \rightarrow \{0,1\}$)
- Each tagging takes input a feature vector

Features for sequence labeling

- Words
  - Current word (essentially like a learned dictionary)
  - Previous/next word (context)
- Other kinds of inferred linguistic classification
  - Part-of-speech tags
- Label context
  - Previous labels

Features: Word substrings



Features: Word shapes

- Word Shapes (Collins)
  - Map words to simplified representation that encodes attributes such as length, capitalization, numerals, Greek letters, internal punctuation, etc.

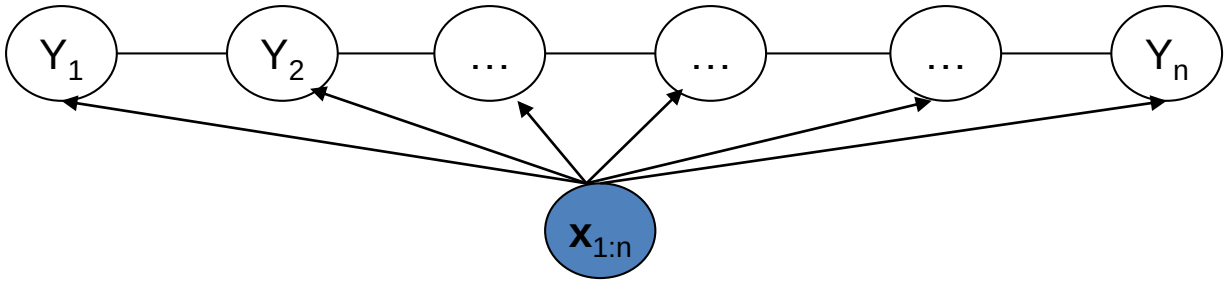
Varicella-zoster	Xx-xxx
mRNA	xXXX
CPA1	XXXd

- Common: all prefixes/suffixes of length ≤ 4

Other Features

- N-gram: Unigram, bigram and trigram token sequences in the context window of the current token
- Part-of-Speech: POS tags of the context words
- Gazetteers: person names, organizations, countries and cities, titles, idioms, etc.
- Word clusters: to reduce sparsity, using word clusters such as Brown clusters (Brown et al., 1992)
- Case and Shape: Capitalization and morphology analysis based features
- Chunking: NP and VP Chunking tags
- Global feature: Sentence level and document level features. For example, whether the token is in the first sentence of a document
- Conjunction: Conjunctions of various features

CRFs (Linear Chain CRF) //Bigram



$$P(\underset{\rho}{y} \mid \underset{\rho}{x}) = \frac{1}{Z(\underset{\rho}{x})} \prod_{t=1}^{T+1} s(y_t, y_{t-1}, \underset{\rho}{x}) = \frac{1}{Z(\underset{\rho}{x})} \prod_{t=1}^{T+1} \exp(\underset{\rho}{w}.\underset{\rho}{\phi}(y_t, y_{t-1}, \underset{\rho}{x}, t)$$

$$Z(\underset{\rho}{x}) = \sum_{\underset{\rho}{y}} \left(\prod_{t=1}^{T+1} \exp(\underset{\rho}{w}.\underset{\rho}{\phi}(y_t, y_{t-1}, \underset{\rho}{x}, t) \right)$$

Decoding (Vitterbi)

- Given learned CRF model compute sequence tags

$$\vec{y}^* = \arg\max_{\vec{y}} \frac{1}{Z(\vec{x})} \prod_{t=1}^{T+1} \exp(\vec{w}.\vec{\phi}(y_t, y_{t-1}, \vec{x}, t) = \arg\max_{\vec{y}} \prod_{t=1}^{T+1} \exp(\vec{w}.\vec{\phi}(y_t, y_{t-1}, \vec{x}, t$$

$$= \arg\max_{\vec{y}} \sum_{t=1}^{T+1} \vec{w}.\vec{\phi}(y_t, y_{t-1}, \vec{x}, t)$$

- Define $\delta_i(y_i) = \max_{y[1:i-1]} \sum_{t=1}^{i} \vec{w}.\vec{\phi}(y_t, y_{t-1}, \vec{x}, t)$

$$\delta_i(y_i) = \max_{y[1:i-1]} \vec{w}.\vec{\phi}(y_i, y_{i-1}, \vec{x}, t) + \sum_{t=1}^{i-1} \vec{w}.\vec{\phi}(y_t, y_{t-1}, \vec{x}, t)$$

$$= \max_{y_{i-1}} \vec{w}.\vec{\phi}(y_i, y_{i-1}, \vec{x}, t) + \max_{y[1:i-2]} \sum_{t=1}^{i-1} \vec{w}.\vec{\phi}(y_t, y_{t-1}, \vec{x}, t)$$

$$= \max_{y_{i-1}} \vec{w}.\vec{\phi}(y_i, y_{i-1}, \vec{x}, t) + \delta_{i-1}(y_{i-1})$$

Training

- Find weights such that

$$LL(\vec{w}) = \log P_{CRF}(\vec{y} \mid \vec{x}; \vec{w}) - \frac{\lambda}{2} \| \vec{w} \|^2$$

is maximized

CRF Learning

- Follows similar ideas as ME models

$$\frac{\partial LL}{\partial w_j} = \sum_d \left(\sum_{t=1}^{T_d} \phi_j(y_{d,t}, y_{d,t-1}, \vec{x}_d, t) - \sum_{\vec{y}} P(\vec{y} \mid \vec{x}_d) \sum_{t=1}^{T_d} \phi_j(y_{d,t}, y_{d,t-1}, \vec{x}_d, t) \right)$$

empirical count — expected count

Exponential sum over all label sequences!

$$\sum_{\vec{y}} P(\vec{y} \mid \vec{x}_d) \sum_{t=1}^{T_d} \phi_j(y_{d,t}, y_{d,t-1}, \vec{x}_d, t) = \sum_{t=1}^{T_d} \left(\sum_{\vec{y}_d} P(\vec{y}_d \mid \vec{x}_d) \phi_j(y_{d,t}, y_{d,t-1}, \vec{x}_d, t) \right)$$

$$= \sum_{t=1}^{T_d} \left(\sum_{y_{d,t-1}, y_{d,t}} P(y_{d,t-1}, y_{d,t} \mid \vec{x}_d) \phi_j(y_{d,t}, y_{d,t-1}, \vec{x}_d, t) \right)$$

Expectation over the corresponding marginal of neighboring nodes -- can be computed efficiently using sum-product algo

34

CRFs: some empirical results

- Parts of Speech tagging

model	*error*	*oov error*
HMM	5.69%	45.99%
MEMM	6.37%	54.61%
CRF	5.55%	48.05%
MEMM$^+$	4.81%	26.99%
CRF$^+$	4.27%	23.76%

$^+$Using spelling features

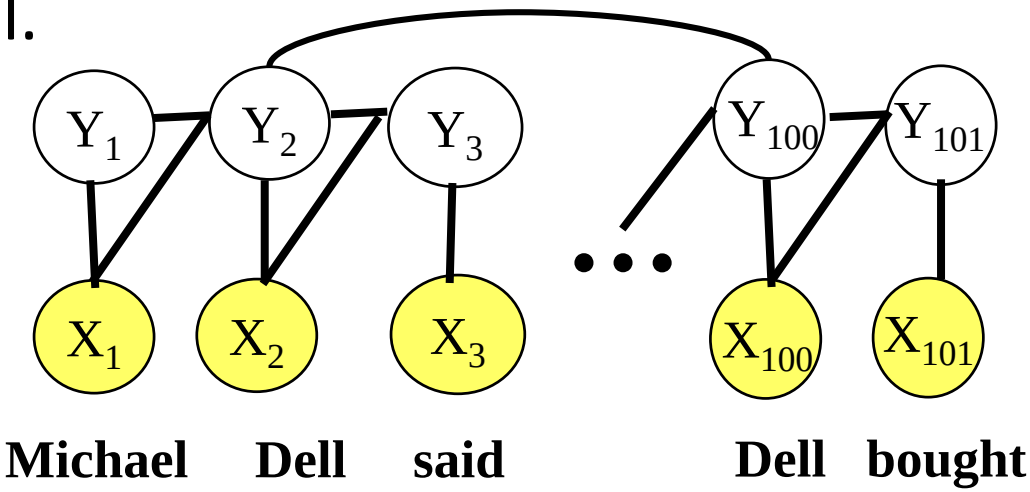
  - Using same set of features: HMM >=< CRF > MEMM
  - Using additional overlapping features: CRF$^+$ > MEMM$^+$ >> HMM

CRF Results

- Experimental results verify that they have superior accuracy on various sequence labeling tasks.
  - Part of Speech tagging
  - Noun phrase chunking
  - Named entity recognition
  - Semantic role labeling
- However, CRFs are much slower to train and do not scale as well to large amounts of training data.
  - Training for POS on full Penn Treebank (~1M words) currently takes “over a week.”

Skip-Chain CRFs

- Can model some long-distance dependencies (i.e. the same word appearing in different parts of the text) by including long-distance edges in the Markov model.



- Additional links make exact inference intractable, so must resort to approximate inference to try to find the most probable labeling.

Linear-chain CRF

$$P(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} \phi_t(y_t, y_{t-1}, x)$$

$$\phi_t(y_t, y_{t-1}, x) = exp(\sum_k (\lambda_k f_k(y_t, y_{t-1}, x, t)))$$

Skip-chain CRF

$$P(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} \phi_t(y_t, y_{t-1}, x) \prod_{(u,v)\in D} \phi_{uv}(y_u, y_v, x)$$

$$\phi_t(y_t, y_{t-1}, x) = exp(\sum_k (\lambda_k f_k(y_t, y_{t-1}, x, t)))$$

$$\phi_{uv}(y_u, y_v, x) = exp(\sum_k (\lambda_{2k} f_{2k}(y_u, y_v, x, u, v)))$$

HMMs vs. CRFs

	HMM (generative)	CRF (discriminative)
Marginal, or Language model: P(sentence)	**Forward algorithm** or **Backward algorithm**, linear in length of sentence	**Can't do it.**
Find optimal label sequence	**Viterbi**, Linear in length of sentence	**Viterbi**, Linear in length of sentence
Supervised parameter estimation	**Bayesian learning**, Easy and fast	**Convex optimization**, Can be quite slow
Unsupervised parameter estimation	**Baum-Welch** (non-convex optimization), Slow but doable	**Very difficult, and requires making extra assumptions.**
Feature functions	Parents and children in the graph →Restrictive!	**Arbitrary functions of a latent state and any portion of the observed nodes**

Summary

- Conditional Random Fields are undirected discriminative models
- De-facto standard for most NLP problems

- Inference for 1-D chain CRFs is exact
  - Same as Max-product or Viterbi decoding
- Learning also is exact
  - globally optimum parameters can be learned
  - Requires using sum-product or forward-backward algorithm

- CRFs involving arbitrary graph structure are intractable in general
  - Skip-chain CRFs improve results on IE.
  - Inference and learning require approximation techniques
    - MCMC sampling
    - Variational methods
    - Loopy BP

Non-local features & Knowledge for NER

Non-local Features

- Identical tokens should have identical label assignments
  - one tag per discourse!

- Counterexample
  - “Australia” (LOC)
  - “The Bank of Australia” (ORG)

- Approaches (suitable for greedy/beam search decoding)
  - Context Aggregation
  - Two-stage Prediction Aggregation
  - Extended Prediction History

Algorithms

Algorithm	Baseline system	Final System
Greedy	83.29	90.57
Beam size=10	83.38	90.67
Beam size=100	83.38	90.67
Viterbi	83.71	N/A

- Viterbi can't be used with non-local features

Context Aggregation & Two-Stage Prediction

- Augment history of a token by
  - aggregating contexts from all occurences of a word
- May result in excessive number of features
- Two-stage prediction
  - use a baseline NER system for first level predictions
  - use prev predictions as features for final prediction

Not All Mentions Made Equal

- Start of a document more important. Why?
  - Often full name mentioned
  - Match gazetteers better
- Introduce prediction-history feature
  - Aggregate feature counting number of times past tokens (same word) were given a certain tag
  - Left to right decoding

Experiments

	Component	CoNLL03 Test data	CoNLL03 Dev data	MUC7 Dev	MUC7 Test	Web pages
1)	Baseline	83.65	89.25	74.72	71.28	71.41
2)	(1) + Context Aggregation	85.40	89.99	**79.16**	71.53	70.76
3)	(1) + Extended Prediction History	**85.57**	**90.97**	78.56	**74.27**	72.19
4)	(1)+ Two-stage Prediction Aggregation	85.01	89.97	75.48	72.16	**72.72**
5)	All Non-local Features (1-4)	86.53	90.69	81.41	73.61	71.21

Knowledge-based NER

- Is Machine Learning necessary?
  - Just do dictionary lookup
    - only 71.91 F1 on CONLL'03

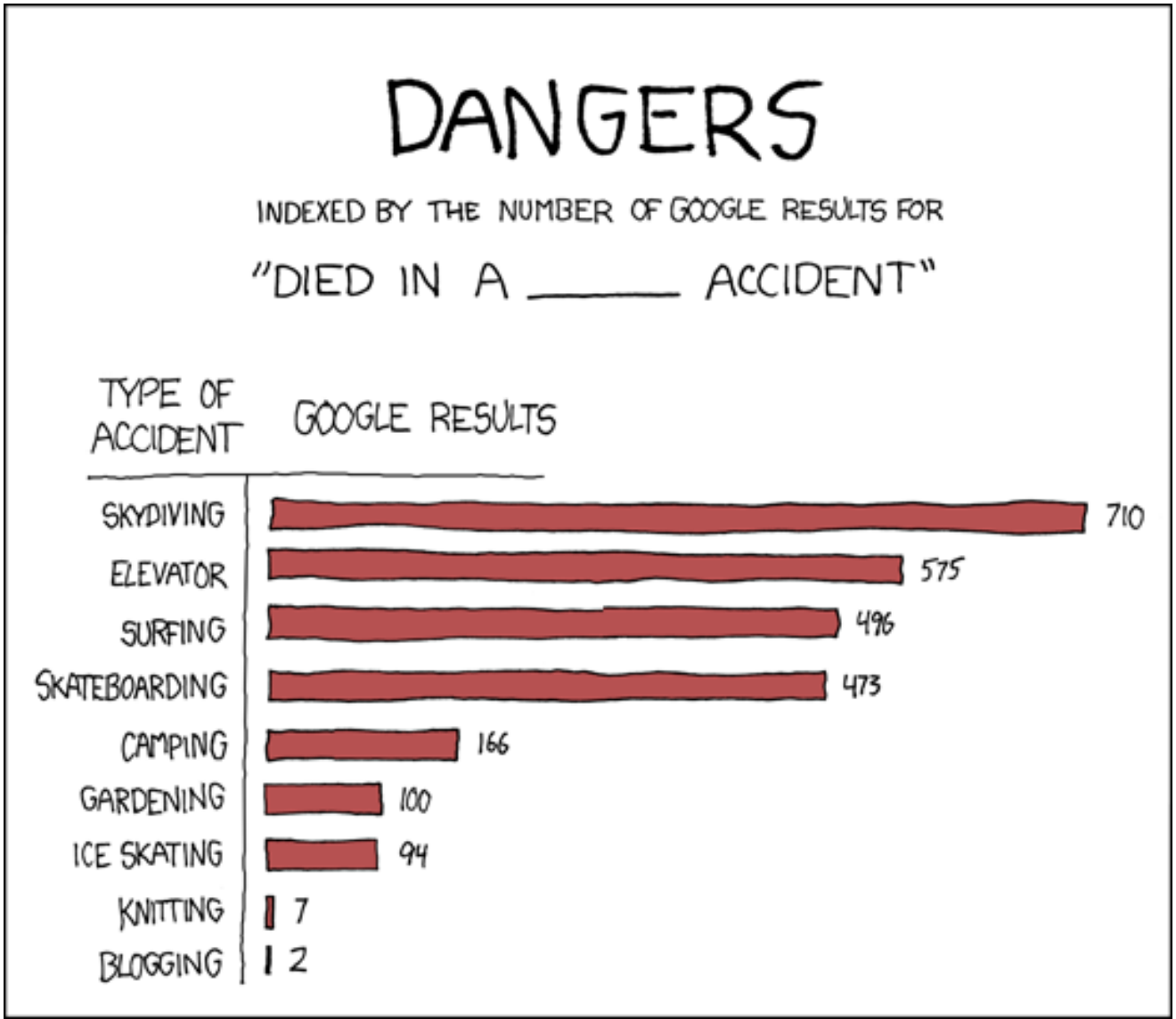
- Use of gazetteers valuable for ML algorithms
  - use existing gazetteers (loc, census data, etc)
  - mine gazetteers from Wikipedia/Freebase
  - auto-construct gazetteers using Web search
  - matches against each gazetteer a different feature

Obtaining Gazetteers Automatically

- Data is Power
  - Web is one of the largest text corpora: however, web search is slooooow (if you have a million queries).
- N-gram data: compressed version of the web
  - Already proven to be useful for language modeling

- Patterns over n-grams
  - To autoconstruct gazetteers

Example: Counts on the Web



Example: Counts on N-grams

died in (a|an) ___ accident

car 13966, automobile 2954, road 1892, auto 1650, traffic 1549, tragic 1480, motorcycle 1399, boating 823, freak 733, drowning 438, vehicle 417, hunting 304, helicopter 289, skiing 281, mining 254, train 250 airplane 236, plane 234, climbing 231, bus 208, motor 198, industrial 187, swimming 180, training 170, motorbike 155, aircraft 152, terrible 137, riding 136, bicycle 132, diving 127, tractor 115, construction 111, farming 107, horrible 105, one-car 104, flying 103, hit-and-run 99, similar 89, racing 89, hiking 89, truck 86, farm 81, bike 78, mine 75, carriage 73, logging 72, unfortunate 71, railroad 71, work-related 70, snowmobile 70, mysterious 68, fishing 67, shooting 66, mountaineering 66, highway 66, single-car 63, cycling 62, air 59, boat 59, horrific 56, sailing 55, fatal 55, workplace 50, skydiving 50, rollover 50, one-vehicle 48, <UNK> 48, work 47, single-vehicle 47, vehicular 45, kayaking 43, surfing 42, automobile 41, car 40, electrical 39, ATV 39, railway 38, Humvee 38, skating 35, hang-gliding 35, canoeing 35, 0000 35, shuttle 34, parachuting 34, jeep 34, ski 33, bulldozer 31, aviation 30, van 30, bizarre 30, wagon 27, two-vehicle 27, street 27, glider 26, " 25, sawmill 25, horse 25, bomb-making 25, bicycling 25, auto 25, alcohol-related 24, snowboarding 24, motoring 24, early-morning 24, trucking 23, elevator 22, horse-riding 22, fire 22, two-car 21, strange 20, mountain-climbing 20, drunk-driving 20, gun 19, rail 18, snowmobiling 17, mill 17, forklift 17, biking 17, river 16, motorcyle 16, lab 16, gliding 16, bonfire 16, apparent 15, aeroplane 15, testing 15, sledding 15, scuba-diving 15, rock-climbing 15, rafting 15, fiery 15, scooter 14, parachute 14, four-wheeler 14, suspicious 13, rodeo 13, mountain 13, laboratory 13, flight 13, domestic 13, buggy 13, horrific 12, violent 12, trolley 12, three-vehicle 12, tank 12, sudden 12, stupid 12, speedboat 12, single 12, jousting 12, ferry 12, airplane 12, unrelated 11, transporter 11, tram 11, scuba 11, common 11, canoe 11, skateboarding 10, ship 10, paragliding 10, paddock 10, moped 10, factory 10

Experiments (CONLL'03)

- Only gazetteer : 71.91
- ML Baseline : 83.65
- Baseline+Gazetteers : 87.22
- Baseline+Gazetteers+Brown : 88.55
- Baseline+Gazetteers+Brown+Non-local : 90.57

	Component	CoNLL03 Test data	CoNLL03 Dev data	MUC7 Dev	MUC7 Test	Web pages
1)	Baseline	83.65	89.25	74.72	71.28	71.41
2)	(1) + External Knowledge	88.55	92.49	84.50	83.23	74.44
3)	(1) + Non-local	86.53	90.69	81.41	73.61	71.21
4)	**All Features**	**90.57**	**93.50**	**89.19**	**86.15**	**74.53**
5)	All Features (train with dev)	90.80	N/A	89.19	86.15	74.33