Text Mining

February 2019

Part-of-Speech Tagging (POS)

- Task of tagging POS tags (Nouns, Verbs, Adjectives, Adverbs, ...) for words
- POS tags provide lot of information about a word
  - knowing whether a word is **noun** or **verb** gives information bout neighbouring words
  - nouns are preceded by determiners; adjectives and verbs by nouns
  - provide useful features for **named entity recognition**
- Given a word, we assume it can belong to only of the POS tags.
- POS Tagging problem
  - Given a sentence $S = w_1 w_2 .... w_n$ consisting of n words, determine the corresponding tag sequence $P = P_1 P_2 .... P_n$

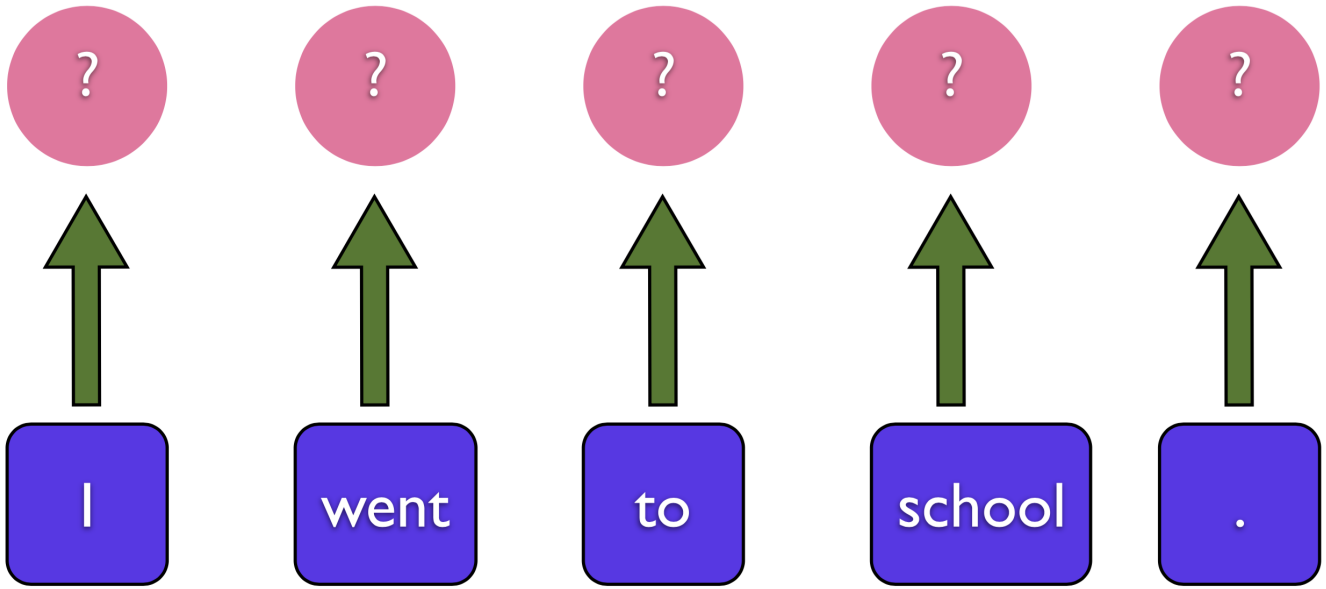
Part-of-Speech Tagging (POS)

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two*	TO	"to"	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential 'there'	*there*	VB	verb base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb gerund	*eating*
JJ	adjective	*yellow*	VBN	verb past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, sing.	*IBM*	$	dollar sign	*$*
NNPS	proper noun, plural	*Carolinas*	#	pound sign	*#*
PDT	predeterminer	*all, both*	"	left quote	' or "
POS	possessive ending	*'s*	"	right quote	' or "
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one's*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

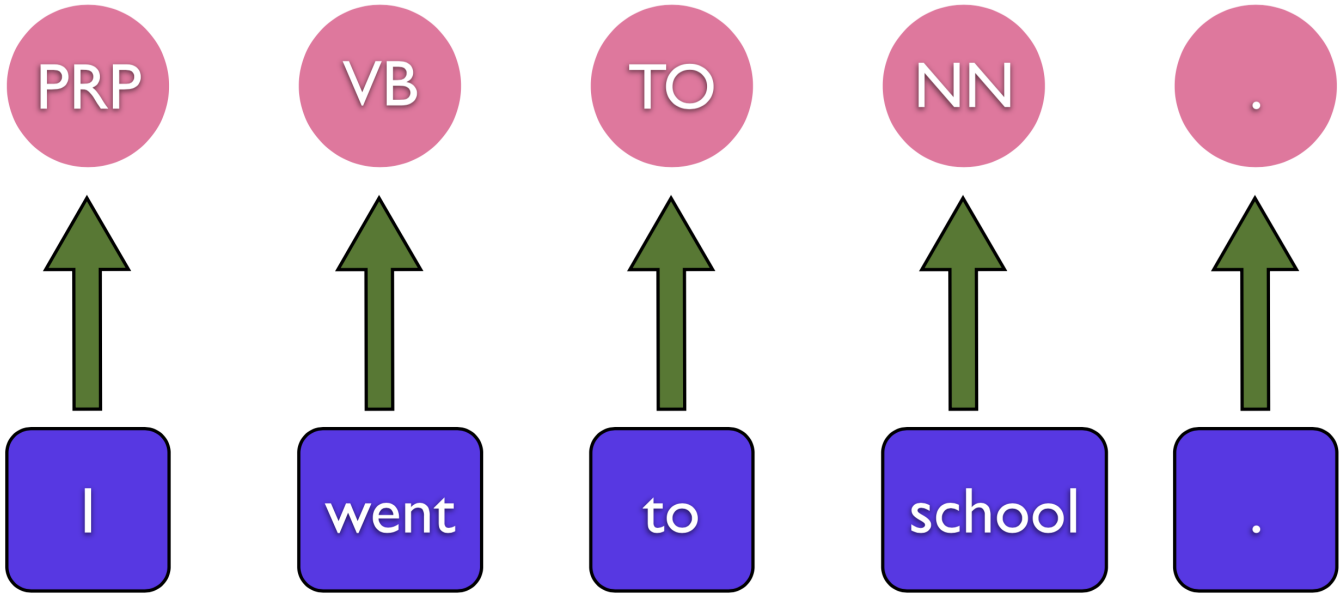
Figure: Penn Treebank POS Tags

Hidden Markov Model (HMM)

Given a sequence of words (observed states) determine a sequence of state transitions (unobserved states)



Hidden Markov Model (HMM)



Markov Chains

- Probabilistic **graphical model** for representing probabilistic assumptions in a graph.

  - $Q = q_1, q_2, ..., q_N$: a set of **states**
  - $A = a_{01}, a_{02}, ...a_{n1}, ..., a_{nn}$: a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, $s.t. \sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
  - q_0, q_{end} : a special *start and end state* which are not associated with observations

Markov Chains

$\pi_1, \pi_2, ..., \pi_N$: an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i.

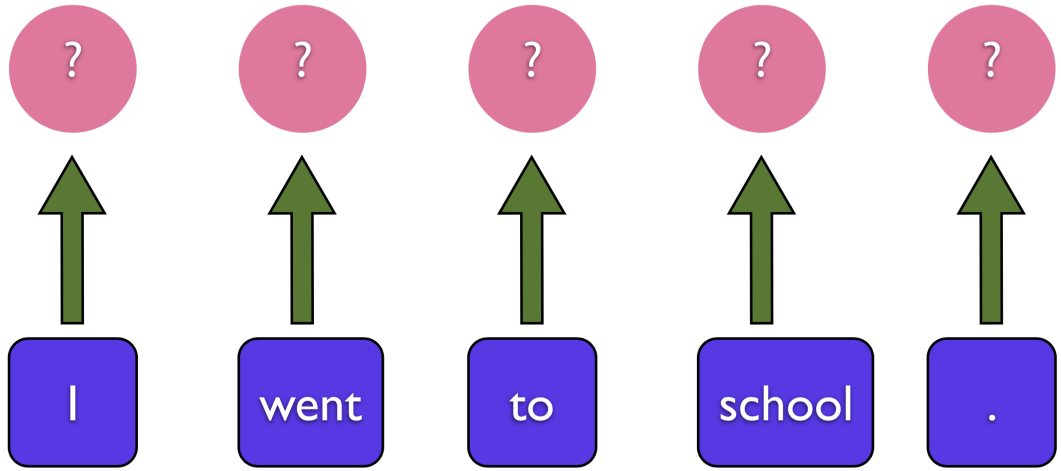
- Markov Assumption:
  $P(q_i|q_1, q_2, ..., q_{i-1}) = P(q_i|q_{i-1})$

- $P(cold\ hot\ cold\ hot) =$
  $P(cold)\ P(hot|cold)\ P(cold|hot)\ P(hot|cold)$
  $= 0.3\,x\,0.2\,x\,0.2\,x\,0.2$
  $= 0.0024$

Hidden Markov Model (HMM)

- Markov chains are useful for observed events
- However, in many cases the events are not observed
  - Example: POS tagging - POS tags are not observed

Given a sequence of words (observed states) determine a sequence of state transitions (unobserved states)



- HMMs allows us to model both *observed events* (words that we see) and *hidden events* (POS tags).

HMM - Definition

$Q = q_1 q_2 \ldots q_N$	a set of **states**
$A = a_{01} a_{02} \ldots a_{n1} \ldots a_{nn}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \ldots o_N$	a set of **observations**, each one drawn from a vocabulary $V = v_1, v_2, ..., v_V$.
$B = b_i(o_t)$	A set of **observation likelihoods:**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state i.
q_0, q_{end}	a special **start and end state** which are not associated with observation.

Markov Assumption: $P(q-1|q_1, ...q_{i-1} = P(q_i|q_{i-1})$

Output Independence Assumption:

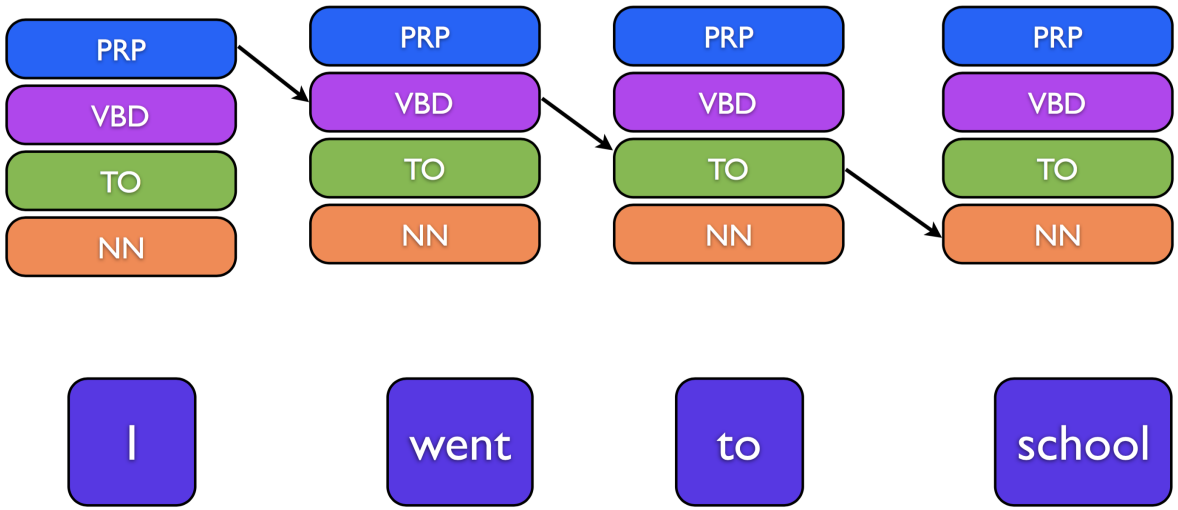
$$P(o_i|q_1, ..., q_i, ..., q_n, o_1, ..., o_i, ..., o_n) = P(o_i|q_i)$$

Three problems of HMM

- **Problem 1 (Decoding)**: Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence S.
- **Problem 2 (Computing Likelihood)**: Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O|\lambda)$.
- **Problem 3 (Learning)** : Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

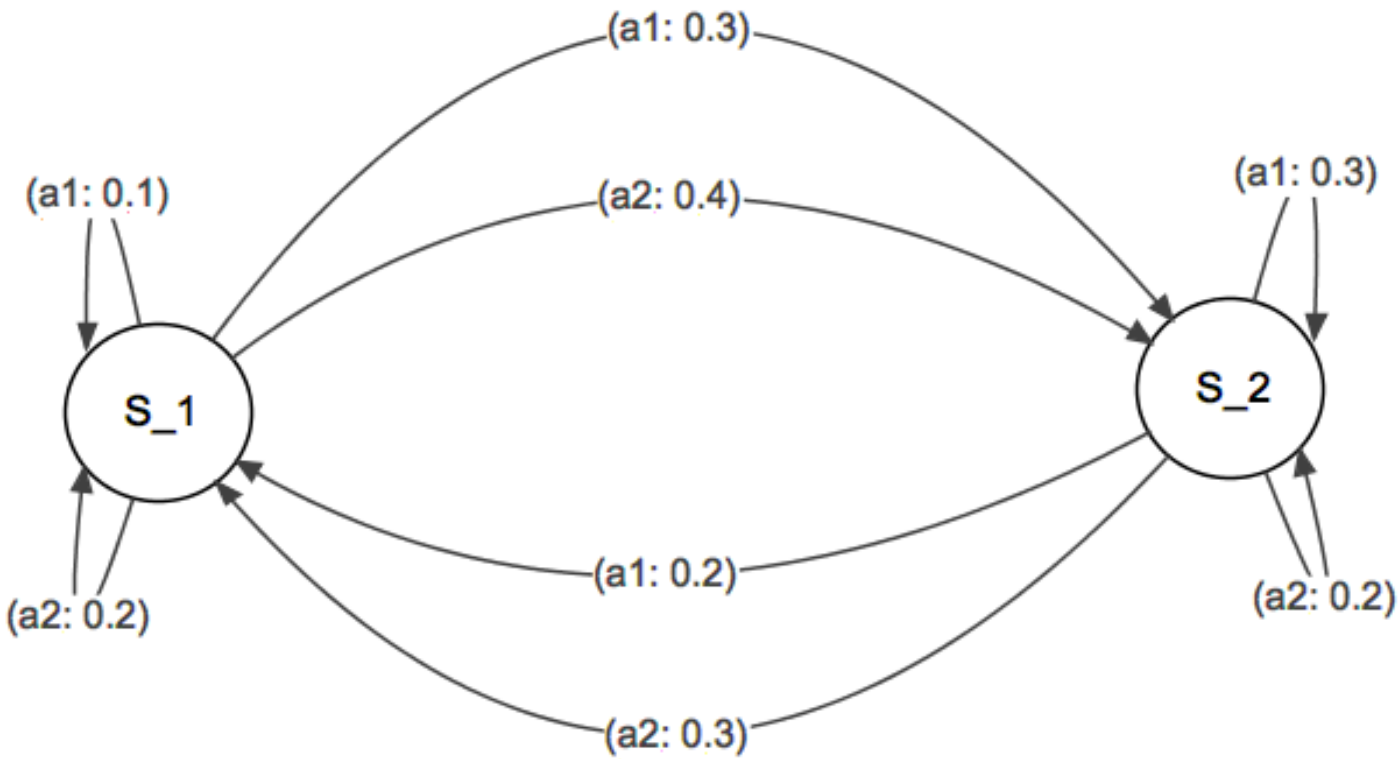
- **Problem 1 (Decoding)**: Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence S.

Why is it difficult?

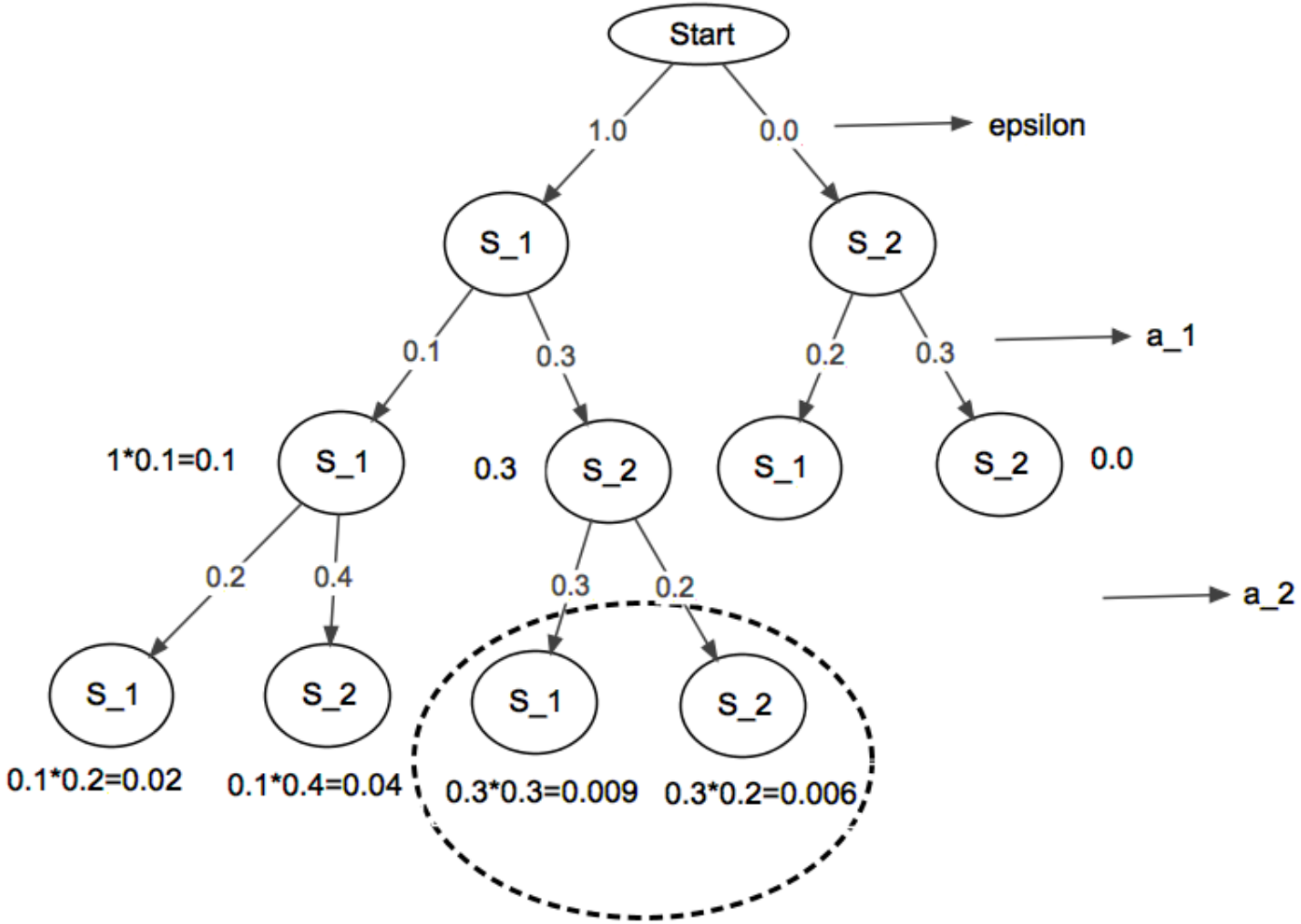


Even if there were only four POS tags, then this is just one of 4x4x4x4 = 256 possible state sequences!

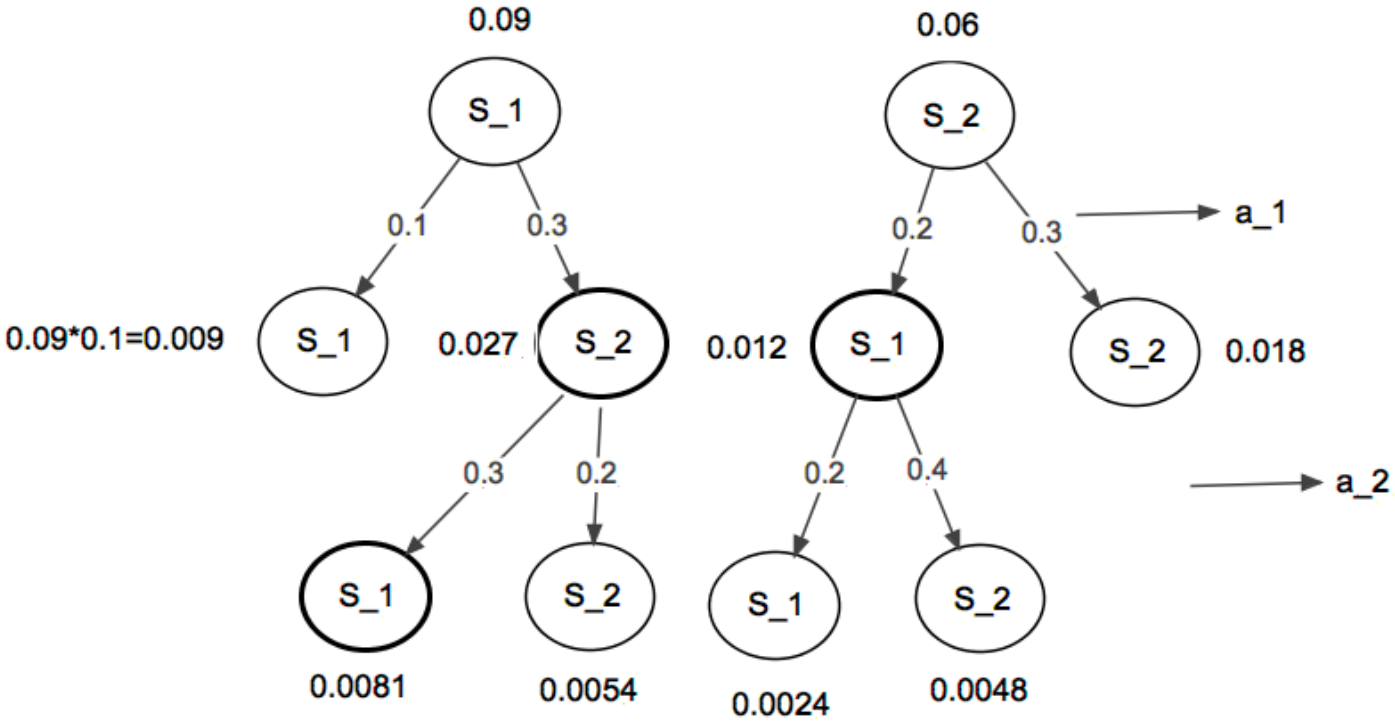
Probabilistic FSM



Probabilistic FSM (contd.)



Probabilistic FSM (contd.)



Tabular Representation of the Tree

	ϵ	a_1	a_2	a_1	a_2
S_1	1.0	(1.0*0.1,0.0*0.2) = (**0.1**,0.0)	(0.02, **0.09**)	(0.009, **0.012**)	(0.0024, **0.0081**)
S_2	0.0	(1.0*0.3,0.0*0.3) = (**0.3**,0.0)	(0.04, **0.06**)	(**0.027**, 0.018)	(0.0048, 0.0054)

- Number of columns = length of observation sequence +1 (ϵ)
- Rows - ending state

HMM - POS Tagging

Goal: choose the most probable tag sequence given the observation sequence of n words $\hat{w}_1^n$

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n)$$

Using Bayes' rule

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} \frac{P(w_1^n | t_1^n) P(t_1^n)}{P(w_1^n)}$$

Simplifying further by dropping the denominator

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(w_1^n | t_1^n) P(t_1^n)$$

HMM - POS Tagging

HMM makes two further assumptions:

1. probability of a word depends only on its tag and is independent of neighbouring words and tags

$$P(w_1^n|t_1^n) \approx \prod_{i=1}^{n} P(w_i|t_i)$$

2. probability of a word depends only on its tag and is independent of neighbouring words and tags

$$P(t_1^n) \approx \prod_{i=1}^{n} P(t_i|t_{i-1})$$

Using these simplifications:

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n|w_1^n) \approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} \overbrace{P(w_i|t_i)}^{\text{emission}} \overbrace{P(t_i|t_{i-1})}^{\text{transition}}$$

HMM - POS Tagging

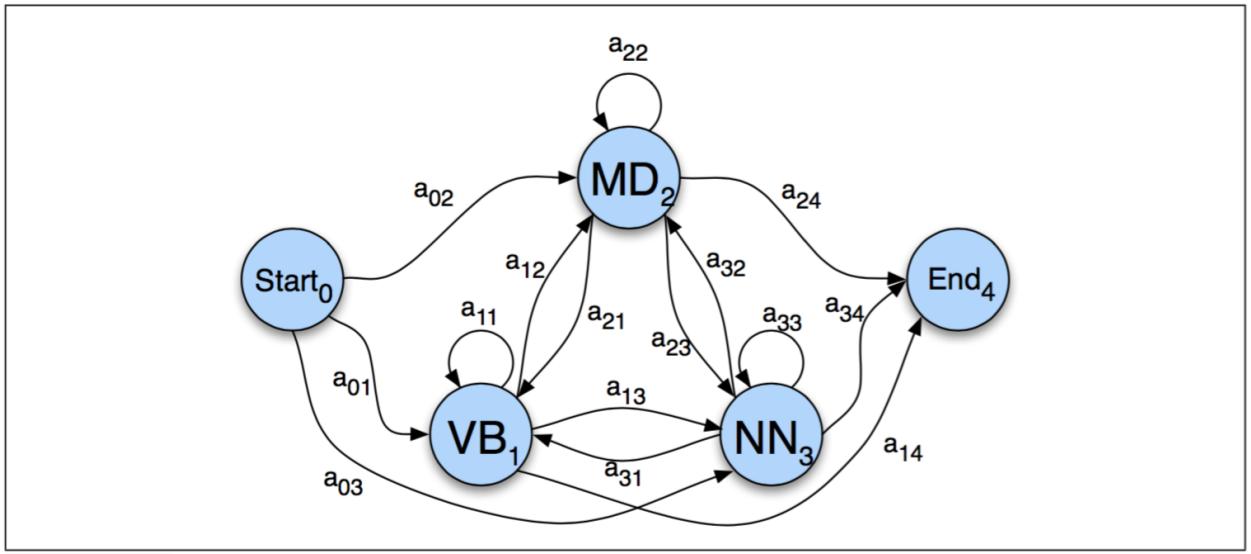


Figure: Markov chain corresponding to the hidden states of HMM. The transition probabilities A are used to compute the prior probability.

HMM - POS Tagging

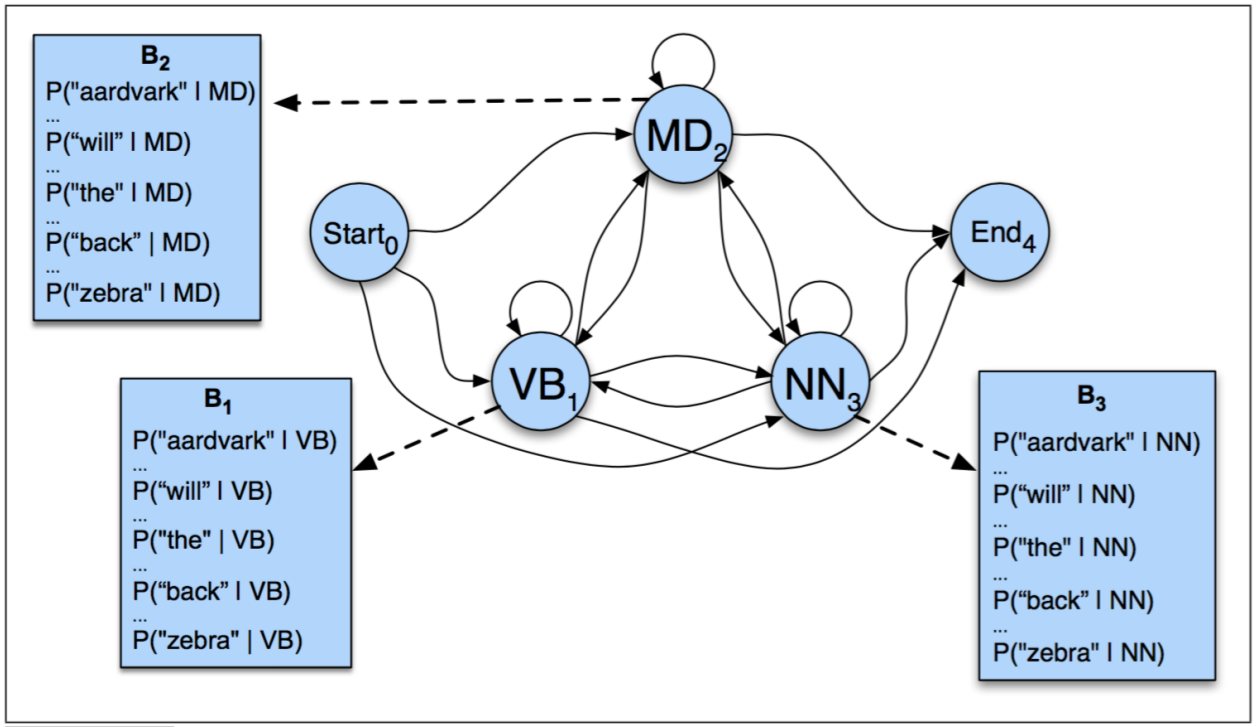


Figure: Observation likelihoods B for the HMM.

Viterbi Algorithm - Pseudocode

function VITERBI(*observations* of len T,*state-graph*) **returns** *best-path*

num-states ← NUM-OF-STATES(*state-graph*)
Create a path probability matrix *viterbi[num-states+2,T+2]*
viterbi[0,0] ← 1.0
for each time step t **from** 1 **to** T **do**
  for each state s **from** 1 **to** *num-states* **do**
    $viterbi[s,t] \leftarrow \max_{1 \leq s' \leq num\text{-}states} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$
    $backpointer[s,t] \leftarrow \operatorname{argmax}_{1 \leq s' \leq num\text{-}states} viterbi[s',t-1] * a_{s',s}$
Backtrace from highest probability state in final column of *viterbi[]* and return path

Figure 6.10 Viterbi algorithm for finding optimal sequence of tags. Given an observation sequence and an HMM $\lambda = (A,B)$, the algorithm returns the state-path through the HMM which assigns maximum likelihood to the observation sequence. Note that states 0 and N+1 are non-emitting *start* and *end* states.

POS Tagging - Example

- Janet will back the bill
- Janet/NNP will/MD back/VB the/DT bill/NN

	NNP	**MD**	**VB**	**JJ**	**NN**	**RB**	**DT**
$<s>$	0.2767	0.0006	0.0031	0.0453	0.0449	0.0510	0.2026
NNP	0.3777	0.0110	0.0009	0.0084	0.0584	0.0090	0.0025
MD	0.0008	0.0002	0.7968	0.0005	0.0008	0.1698	0.0041
VB	0.0322	0.0005	0.0050	0.0837	0.0615	0.0514	0.2231
JJ	0.0366	0.0004	0.0001	0.0733	0.4509	0.0036	0.0036
NN	0.0096	0.0176	0.0014	0.0086	0.1216	0.0177	0.0068
RB	0.0068	0.0102	0.1011	0.1012	0.0120	0.0728	0.0479
DT	0.1147	0.0021	0.0002	0.2157	0.4744	0.0102	0.0017

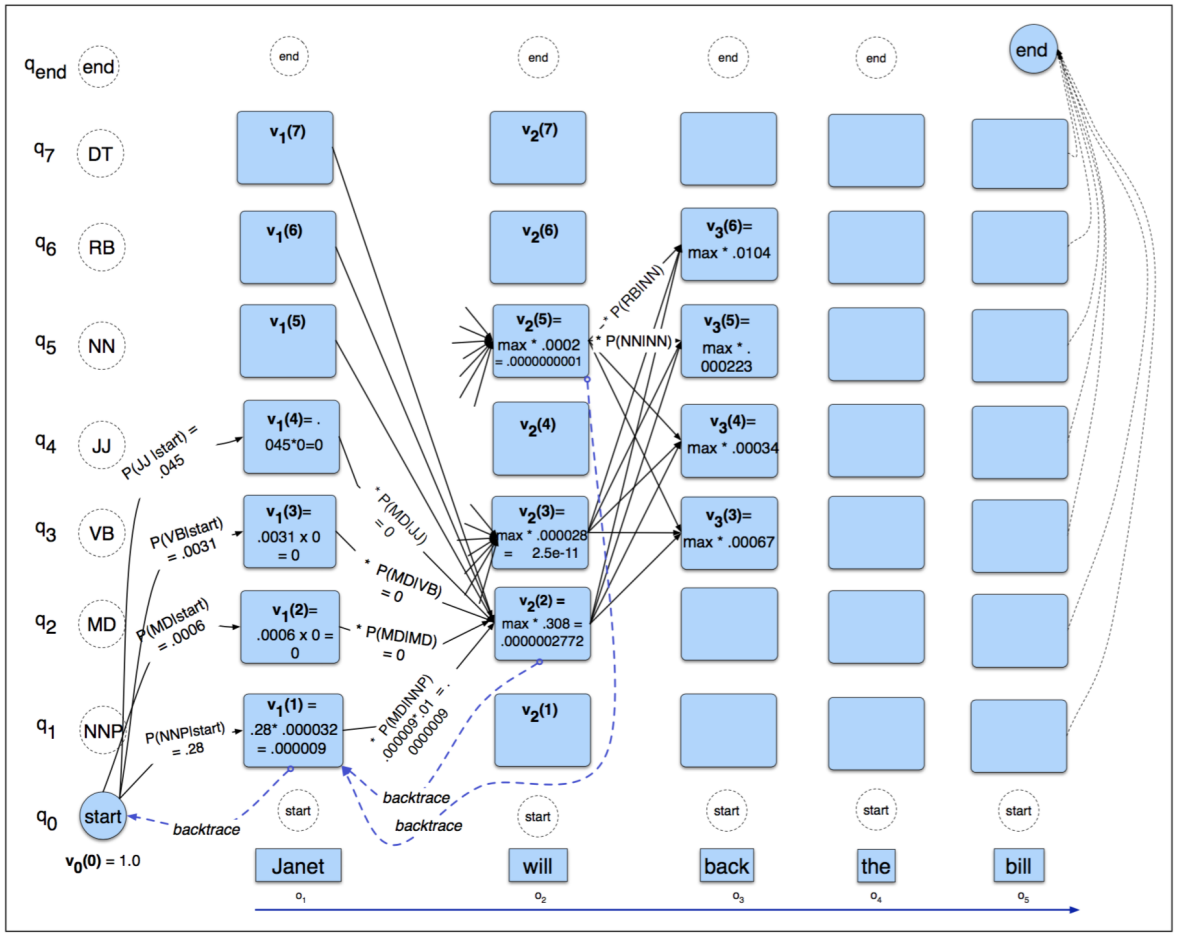
POS Tagging - Example

- Janet will back the bill
- Janet/NNP will/MD back/VB the/DT bill/NN

	Janet	will	back	the	bill
NNP	0.000032	0	0	0.000048	0
MD	0	0.308431	0	0	0
VB	0	0.000028	0.000672	0	0.000028
JJ	0	0	0.000340	0.000097	0
NN	0	0.000200	0.000223	0.000006	0.002337
RB	0	0	0.010446	0	0
DT	0	0	0	0.506099	0

POS Tagging - Example

- Janet will back the bill
- Janet/NNP will/MD back/VB the/DT bill/NN



- Relation Extraction
  - What it is?
- Relation Extraction Methods
  - Hand-built patterns
  - Bootstrapping methods
  - Supervised learning methods
  - Distant supervision

Relation Extraction - Example

CHICAGO (AP) — Citing high fuel prices, United Airlines said Friday it has increased fares by $6 per round trip on flights to some cities also served by lower-cost carriers. American Airlines, a unit of AMR, immediately matched the move, spokesman Tim Wagner said. United, a unit of UAL, said the increase took effect Thursday night and applies to most routes where it competes against discount carriers, such as Chicago to Dallas and Atlanta and Denver to San Francisco, Los Angeles and New York.

Question: What relations should we extract?

slide adapted from Luke Zettlemoyer

Relation Extraction - Example

CHICAGO (AP) — Citing high fuel prices, United Airlines said Friday it has increased fares by $6 per round trip on flights to some cities also served by lower-cost carriers. American Airlines, a unit of AMR, immediately matched the move, spokesman Tim Wagner said. United, a unit of UAL, said the increase took effect Thursday night and applies to most routes where it competes against discount carriers, such as Chicago to Dallas and Atlanta and Denver to San Francisco, Los Angeles and New York.

Subject	Relation	Object
American Airlines	subsidiary	AMR
Tim Wagner	employee	American Airlines
United Airlines	subsidiary	UAL

slide adapted from Luke Zettlemoyer

Relation Types

For generic news texts ...

Relations		Examples	Types
Affiliations			
	Personal	*married to, mother of*	PER → PER
	Organizational	*spokesman for, president of*	PER → ORG
	Artifactual	*owns, invented, produces*	(PER | ORG) → ART
Geospatial			
	Proximity	*near, on outskirts*	LOC → LOC
	Directional	*southeast of*	LOC → LOC
Part-Of			
	Organizational	*a unit of, parent of*	ORG → ORG
	Political	*annexed, acquired*	GPE → GPE

slide adapted from Jim Martin

Relation Types - ACE 2003

ROLE: relates a person to an organization or a geopolitical entity
subtypes: member, owner, affiliate, client, citizen

PART: generalized containment
subtypes: subsidiary, physical part-of, set membership

AT: permanent and transient locations
subtypes: located, based-in, residence

SOCIAL: social relations among persons
subtypes: parent, sibling, spouse, grandparent, associate

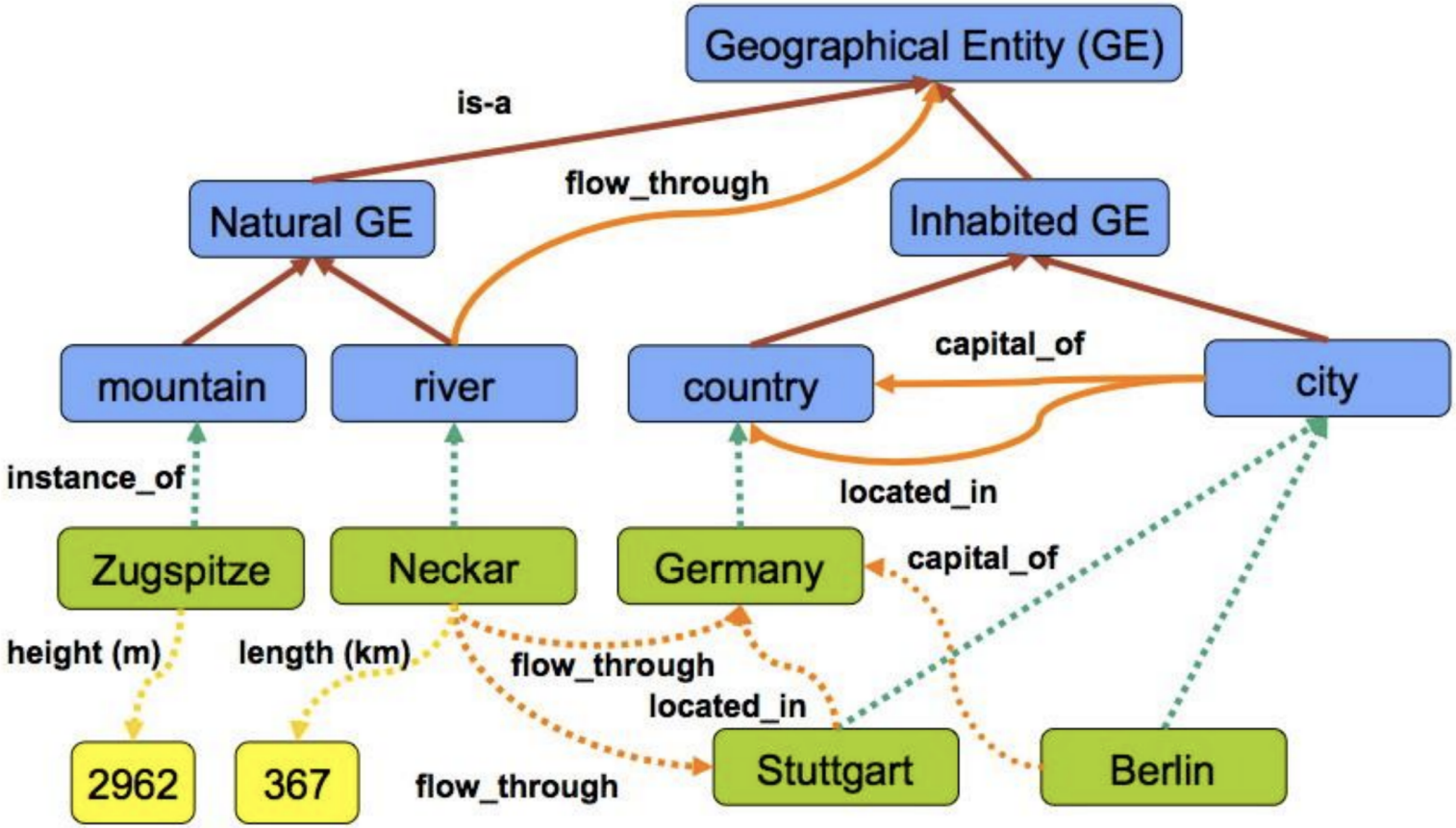
slide adapted from Doug Appelt

Relation Types - Freebase

23 Million Entities, thousands of relations

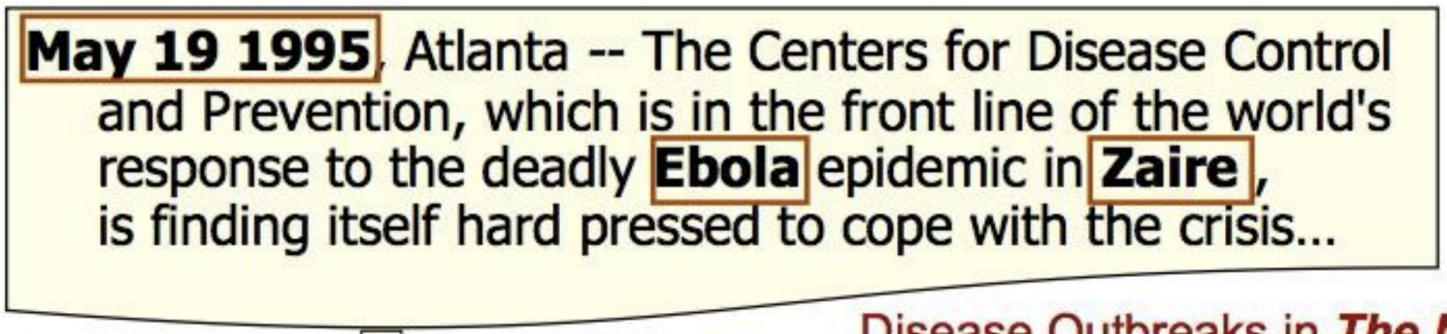
Relation name	Size	Example
/people/person/nationality	281,107	John Dugard, South Africa
/location/location/contains	253,223	Belgium, Nijlen
/people/person/profession	208,888	Dusa McDuff, Mathematician
/people/person/place_of_birth	105,799	Edwin Hubble, Marshfield
/dining/restaurant/cuisine	86,213	MacAyo's Mexican Kitchen, Mexican
/business/business_chain/location	66,529	Apple Inc., Apple Inc., South Park, NC
/biology/organism_classification_rank	42,806	Scorpaeniformes, Order
/film/film/genre	40,658	Where the Sidewalk Ends, Film noir
/film/film/language	31,103	Enter the Phoenix, Cantonese
/biology/organism_higher_classification	30,052	Calopteryx, Calopterygidae
/film/film/country	27,217	Turtle Diary, United States
/film/writer/film	23,856	Irving Shulman, Rebel Without a Cause
/film/director/film	23,539	Michael Mann, Collateral
/film/producer/film	22,079	Diane Eskenazi, Aladdin
/people/deceased_person/place_of_death	18,814	John W. Kern, Asheville
/music/artist/origin	18,619	The Octopus Project, Austin
/people/person/religion	17,582	Joseph Chartrand, Catholicism
/book/author/works_written	17,278	Paul Auster, Travels in the Scriptorium
/soccer/football_position/players	17,244	Midfielder, Chen Tao
/people/deceased_person/cause_of_death	16,709	Richard Daintree, Tuberculosis
/book/book/genre	16,431	Pony Soldiers, Science fiction
/film/film/music	14,070	Stavisky, Stephen Sondheim
/business/company/industry	13,805	ATS Medical, Health care

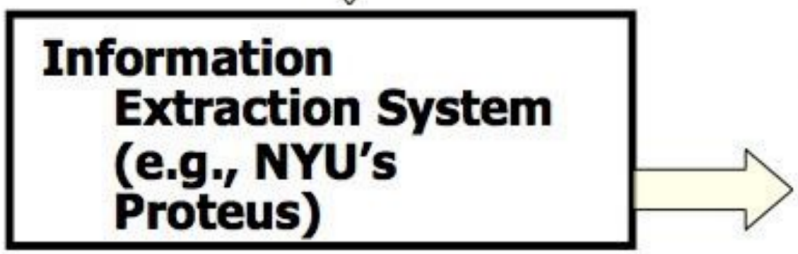
Relation Types - Geospatial



slide adapted from Paul Buitelaar

Relation Types - Disease Outbreaks





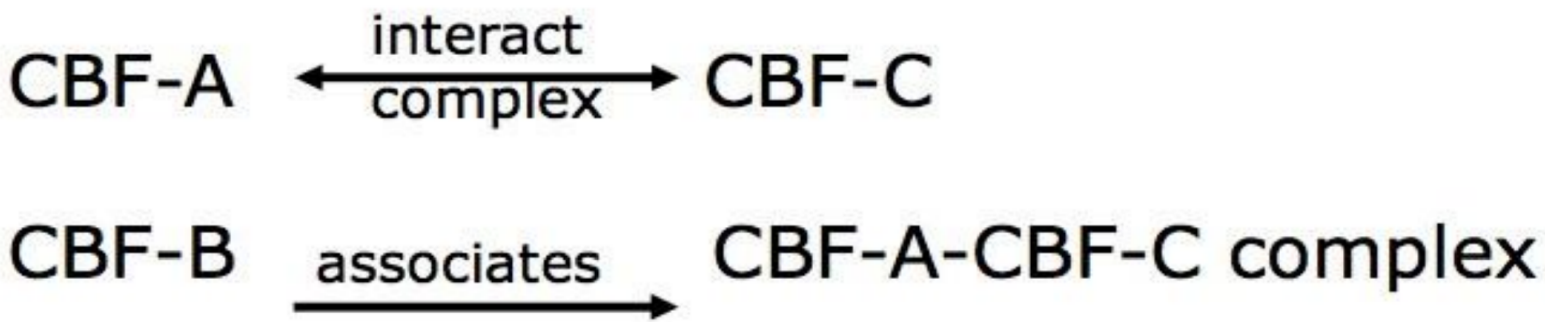
Disease Outbreaks in *The New York Times*

Date	*Disease Name*	*Location*
Jan. 1995	Malaria	Ethiopia
July 1995	Mad Cow Disease	U.K.
Feb. 1995	Pneumonia	U.S.

slide adapted from Eugene Agichtein

Relation Types - Protein Interactions

„We show that CBF-A and CBF-C interact with each other to form a CBF-A-CBF-C complex and that CBF-B does not interact with CBF-A or CBF-C individually but that it associates with the CBF-A-CBF-C complex."



slide adapted from Rosario & Hearst

Relation Extraction - Need

- NLP applications need word meaning!
  - Question answering
  - Conversational agents
  - Summarization
- One key meaning component: word relations
  - Hyponymy: San Francisco is an instance of a city
  - Antonymy: acidic is the opposite of basic
  - Meronymy: an alternator is a part of a car

slide adapted from Luke Zettlemoyer

Relation Extraction - Need

WordNet is incomplete

Ontological relations are missing for many words:

In WordNet 3.1	Not in WordNet 3.1
insulin progesterone	leptin pregnenolone
combustibility navigability	affordability reusability
HTML	XML
Google, Yahoo	Microsoft, IBM

Esp. for specific domains: restaurants, auto parts, finance

slide adapted from Luke Zettlemoyer

Relation Extraction - Methods

- Hand-built patterns
- Bootstrapping methods
- Supervised learning methods
- Distant supervision method

Hand-built Patterns

What does *Gelidium* mean?

Hand-built Patterns

What does *Gelidium* mean?

What does *Agar is a substance prepared from a mixture of red algae, such as Gelidium, for laboratory or industrial use*

Hearst's Lexico-Syntactic Patterns

Y such as X ((, X)* (, and/or) X)

such Y as X…

X… or other Y

X… and other Y

Y including X…

Y, especially X…

Hearst, 1992. Automatic Acquisition of Hyponyms

Examples of the Hearst's Patterns

Hearst pattern	Example occurrences
X and other Y	...temples, treasuries, and other important civic buildings.
X or other Y	bruises, wounds, broken bones or other injuries...
Y such as X	The bow lute, such as the Bambara ndang...
such Y as X	...such authors as Herrick, Goldsmith, and Shakespeare.
Y including X	...common-law countries, including Canada and England...
Y, especially X	European countries, especially France, England, and Spain...

slide adapted from Luke Zettlemoyer

Problems with Hand-built Patterns

- Requires hand-building patterns for each relation!
  - hard to write; hard to maintain
  - there are zillions of them
  - domain-dependent
- Don't want to do this for all possible relations!
- Plus, we'd like better accuracy
  - Hearst: 66% accuracy on hyponym extraction

slide adapted from Luke Zettlemoyer

Relation Extraction - Methods

- Hand-built patterns
- Bootstrapping methods
- Supervised learning methods
- Distant supervision method

Bootstrapping Approach

- If you don't have enough annotated text to train on …
- But you do have:
  - some seed instances of the relation
  - (or some patterns that work pretty well)
  - and lots & lots of unannotated text (e.g., the web)
- … can you use those seeds to do something useful?
- Bootstrapping can be considered *semi-supervised*

slide adapted from Luke Zettlemoyer

Bootstrapping Example

Seed: (Arthur Conan Doyle, The Adventures of Sherlock Holmes)

A Web crawler finds all documents contain the pair.

slide adapted from Nguyen Bach and Sameer Badaskar

Bootstrapping - Matched Document 1

...

Read The Adventures of Sherlock Holmes by Arthur Conan Doyle online or in you email

...

Extract **tuple**:

[0, Arthur Conan Doyle, The Adventures of Sherlock Holmes, Read, online or, by]

A tuple of 6 elements: [order, author, book, prefix, suffix, middle]

order = 1 if the author string occurs before the book string, = 0 otherwise

prefix and *suffix* are strings contain the 10 characters occurring to the left/right of the match

middle is the string occurring between the author and book

slide adapted from Nguyen Bach and Sameer Badaskar

Bootstrapping - Matched Document 2

...

know that Sir Arthur Conan Doyle wrote The Adventures of Sherlock Holmes, in 1892

...

Extract tuple:

[1, Arthur Conan Doyle, The Adventures of Sherlock Holmes, now that Sir, in 1892, wrote]

slide adapted from Nguyen Bach and Sameer Badaskar

Bootstrapping - Developing Patterns

Extracted list of tuples:

[0, Arthur Conan Doyle, The Adventures of Sherlock Holmes, Read, online or, by]

[1, Arthur Conan Doyle, The Adventures of Sherlock Holmes, now that Sir, in 1892, wrote]

[1, Arthur Conan Doyle, The Adventures of Sherlock Holmes, When Sir, in 1892 he, wrote]

...

Group tuples by matching *order* and *middle* and induce patterns

Induce patterns from group of tuples:

[longest-common-suffix of prefix strings, author, middle, book, longest-common-prefix of suffix strings]

Pattern:

[Sir, Arthur Conan Doyle, wrote, The Adventures of Sherlock Holmes, in 1892]

Pattern with wild card expression:

[Sir, .*?, wrote, .*?, in 1892]

slide adapted from Nguyen Bach and Sameer Badaskar

Bootstrapping - Search for more patterns and new tuples

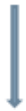
Use the wild card patterns **[Sir, .*?, wrote, .*?, in 1892]**

search the Web to find more documents

...

Sir Arthur Conan Doyle **wrote** Speckled Band **in 1892**, that is around 62 years apart which would make the stories

...



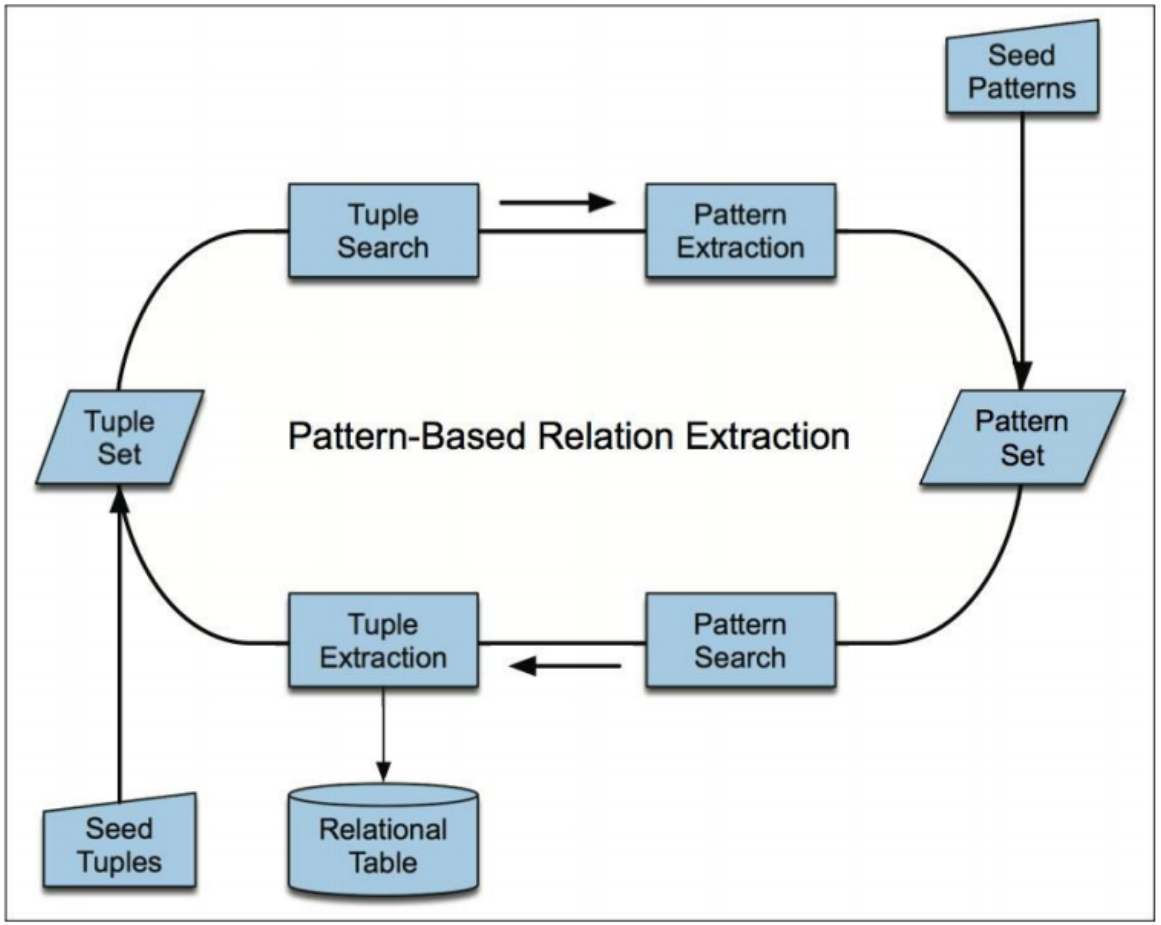
Extract new relations:

(Arthur Conan Doyle, Speckled Band)

Repeat the algorithm with the new relation.

slide adapted from Nguyen Bach and Sameer Badaskar

Bootstrapping System



slide adapted from Jim Martin

Bootstrapping - Problems

- Requires that we have seeds for each relation
  - Sensitive to original set of seeds
- Big problem of semantic drift at each iteration
- Precision tends to be not that high
- Generally have lots of parameters to be tuned
- No probabilistic interpretation
  - Hard to know how confident to be in each result

slide adapted from Luke Zettlemoyer

Next Class

- Relation Extraction Methods
  - Supervised Learning
  - Distant Supervision
  - *n*-ary Relation Extraction
- Question-Answering