Part of Speech Tagging and Hidden Markov Model

Instructor: Wei Xu

Some slides adapted from Brendan O'Connor, Chris Manning, Michael Collins, and Yejin Choi

Where are we going with this?

- Text classification: bags of words
- Language Modeling: n-grams
- Sequence tagging:
  - Parts of Speech
  - Named Entity Recognition
  - Other areas: bioinformatics (gene prediction), etc…

What’s a part-of-speech (POS)?

- Syntax = how words compose to form larger meaning bearing units

- POS = syntactic categories for words (a.k.a word class)
  - You could substitute words within a class and have a syntactically valid sentence

I saw the **dog**

I saw the **cat**

I saw the ___

  - Gives information how words combine into larger phrases

Parts of Speech is an old idea

- Perhaps starting with Aristotle in the West (384–322 BCE), there was the idea of having parts of speech
- Also, Dionysius Thrax of Alexandria (c. 100 BCE)
- 8 main POS: noun, verb, adjective, adverb, preposition, conjunction, pronoun, interjection
- Many more fine grained possibilities

https://www.youtube.com/watch?v=ODGA7ssL-6g&index=1&list=PL6795522EAD6CE2F7

cs.jhu.edu/~jonny/thrax/

Thrax

an extractor for synchronous context-free grammars for machine translation

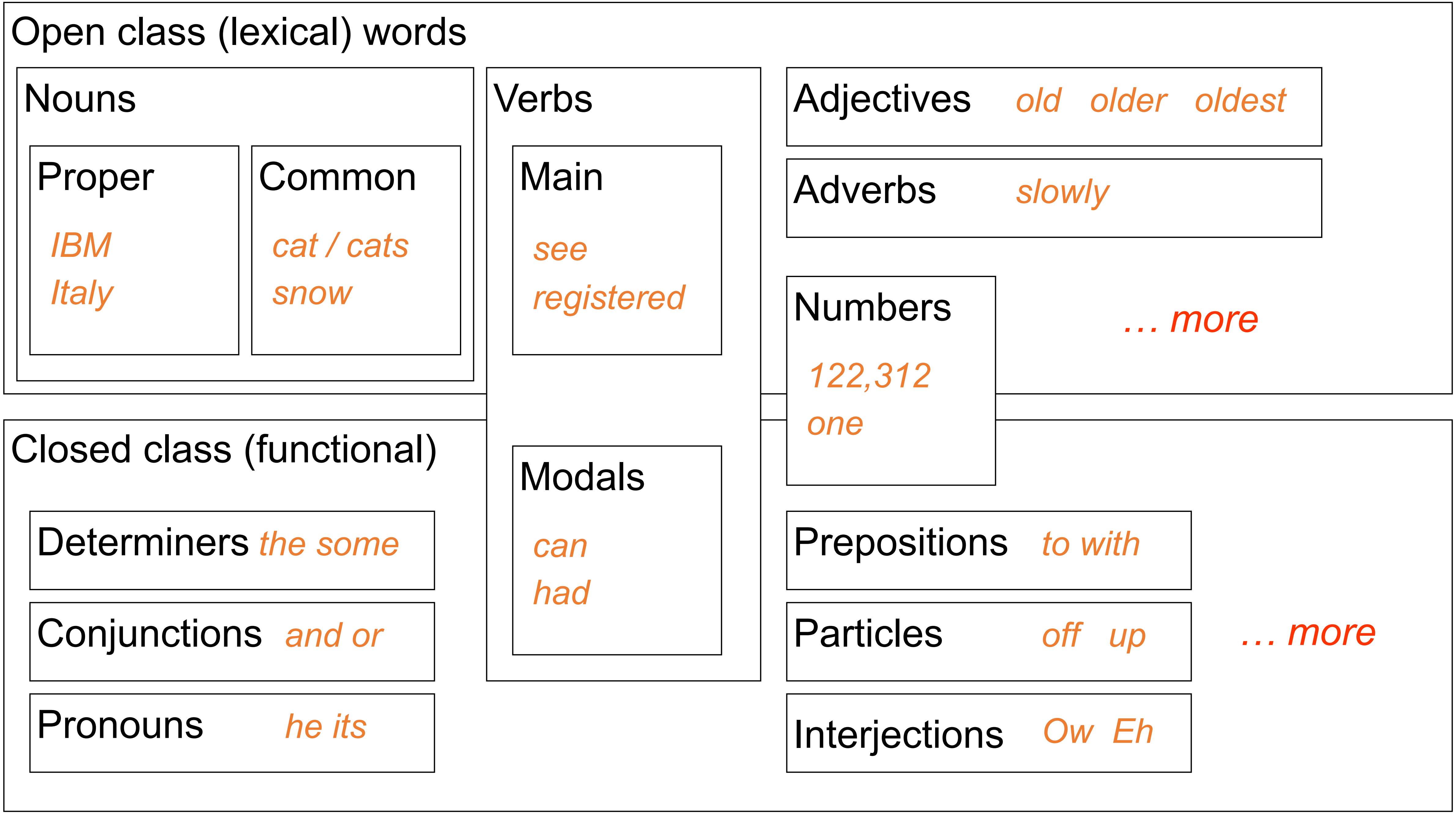
(*Allegory of Grammar* by Laurent de La Hyre)

What it is

As the banner indicates, Thrax is an extractor for synchronous context-free grammars (SCFGs) for use in machine translation (MT). This paper has a nice introduction to the SCFG formalism for translation.

Why it's called what it's called

Thrax is so named in honor of Dionysius Thrax. He's credited with creating the first grammar of Greek, *Art of Grammar*. Since this program is designed to create grammars, we thought it was a clever reference. Plus the name is short and catchy and has no obvious relation to the program's function, which is traditional for UNIX-style program names.


Open class (lexical) words
Nouns
Proper
IBM
Italy
Common
cat / cats
snow
Verbs
Main
see
registered
Modals
can
had
Adjectives old older oldest
Adverbs slowly
Numbers
122,312
one
… more
Closed class (functional)
Determiners the some
Conjunctions and or
Pronouns he its
Prepositions to with
Particles off up
Interjections Ow Eh
… more

Open vs. Closed classes

- Open vs. Closed classes
  - Closed:
    - determiners: ***a*, *an*, *the***
    - pronouns: ***she*, *he*, *I***
    - prepositions: ***on*, *under*, *over*, *near*, *by*, …**
    - Q: why called "closed"?
  - Open:
    - Nouns, Verbs, Adjectives, Adverbs.

Many Tagging Standards

- Penn Treebank (45 tags) … this is the most common one
- Brown corpus (85 tags)
- Coarse tagsets
  - Universal POS tags (Petrov et. al. https://github.com/slavpetrov/universal-pos-tags)
  - Motivation: cross-linguistic regularities

Penn Treebank POS

- 45 possible tags
- 34 pages of tagging guidelines

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	+,%, &
CD	Cardinal number	*one, two, three*	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, singular	*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*			

https://catalog.ldc.upenn.edu/docs/LDC99T42/tagguid1.pdf

Ambiguity in POS Tagging

- Words often have more than one POS: *back*
  - The *back* door = JJ
  - On my *back* = NN
  - Win the voters *back* = RB
  - Promised to *back* the bill = VB
- The POS tagging problem is to determine the POS tag for a particular instance of a word.

Exercise

POS Tagging

- Input: Plays well with others
- Ambiguity: NNS/VBZ UH/JJ/NN/RB IN NNS
- Output: Plays/VBZ well/RB with/IN others/NNS

Penn Treebank POS tags

POS Tagging Performance

- How many tags are correct? (Tag Accuracy)
  - About 97% currently
  - But baseline is already 90%
    - Baseline is performance of stupidest possible method
      - Tag every word with its most frequent tag
      - Tag unknown words as nouns
  - Partly easy because
    - Many words are unambiguous
    - You get points for them (*the*, *a*, etc.) and for punctuation marks!

Deciding on the correct part of speech can be difficult even for people

- “Around” can be a particle, preposition, or adverb

Mrs/NNP Schaefer/NNP never/RB got/VBD around/RP to/TO joining/VBG

All/DT we/PRP gotta/VBN do/VB is/VBZ go/VB around/IN the/DT corner/NN

Chateau/NNP Petrus/NNP costs/VBZ around/RB 250/CD

It's hard for linguists too!

4 Confusing parts of speech

This section discusses parts of speech that are easily confused and gives guidelines on how to tag such cases.

CD or JJ

Number-number combinations should be tagged as adjectives (JJ) if they have the same distribution as adjectives.

EXAMPLES: a 50–3/JJ victory (cf. a handy/JJ victory)

Hyphenated fractions *one-half*, *three-fourths*, *seven-eighths*, *one-and-a-half*, *seven-and-three-eighths* should be tagged as adjectives (JJ) when they are prenominal modifiers, but as adverbs (RB) if they could be replaced by *double* or *twice*.

EXAMPLES: one-half/JJ cup; cf. a full/JJ cup
one-half/RB the amount; cf. twice/RB the amount; double/RB the amount

https://catalog.ldc.upenn.edu/docs/LDC99T42/tagguid1.pdf

How difficult is POS tagging?

- About 11% of the **word types** in the Brown corpus are ambiguous with regard to part of speech
- But they tend to be very common words. E.g., *that*
  - I know *that* he is honest = IN
  - Yes, *that* play was nice = DT
  - You can't go *that* far = RB
- 40% of the **word tokens** are ambiguous

Token vs. Type

Token is instance or individual occurrence of a type.

Why POS Tagging?

- Useful in and of itself (more than you'd think)
  - Text-to-speech: record, lead
  - Lemmatization: saw[v] → see, saw[a] → saw
  - Quick-and-dirty NP-chunk detection: grep {JJ | NN}* {NN | NNS}

Quick-and-Dirty Noun Phrase Identification

Grammatical structure: Candidate strings are those multi-word noun phrases that are specified by the regular expression $((A \mid N)^+ \mid ((A \mid N)^*(NP)^?)(A \mid N)^*)N$,

Tag Pattern	Example
A N	*linear function*
N N	*regression coefficients*
A A N	*Gaussian random variable*
A N N	*cumulative distribution function*
N A N	*mean squared error*
N N N	*class probability function*
N P N	*degrees of freedom*

Table 5.2 Part of speech tag patterns for collocation filtering. These patterns were used by Justeson and Katz to identify likely collocations among frequently occurring word sequences.

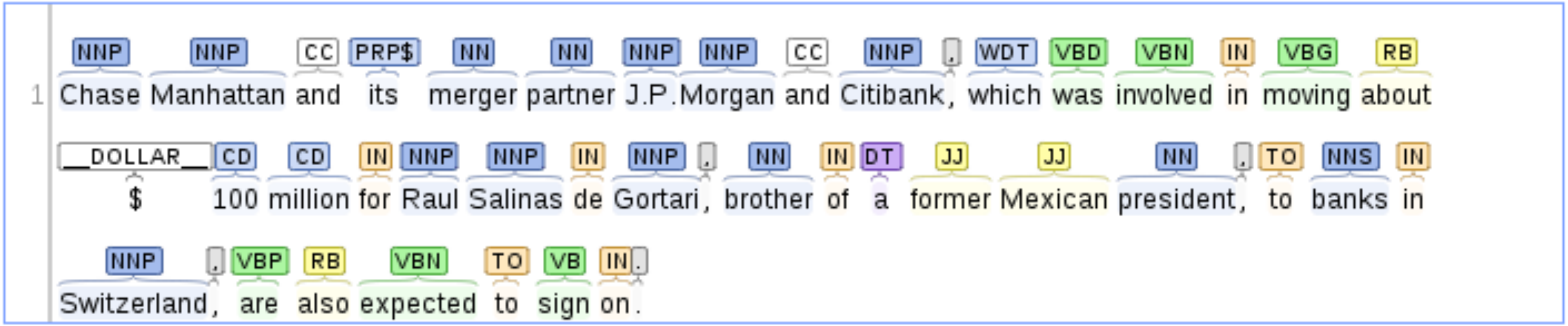
Why POS Tagging?

- Useful in and of itself (more than you'd think)
  - Text-to-speech: record, lead
  - Lemmatization: saw[v] → see, saw[a] → saw
  - Quick-and-dirty NP-chunk detection: grep {JJ | NN}* {NN | NNS}

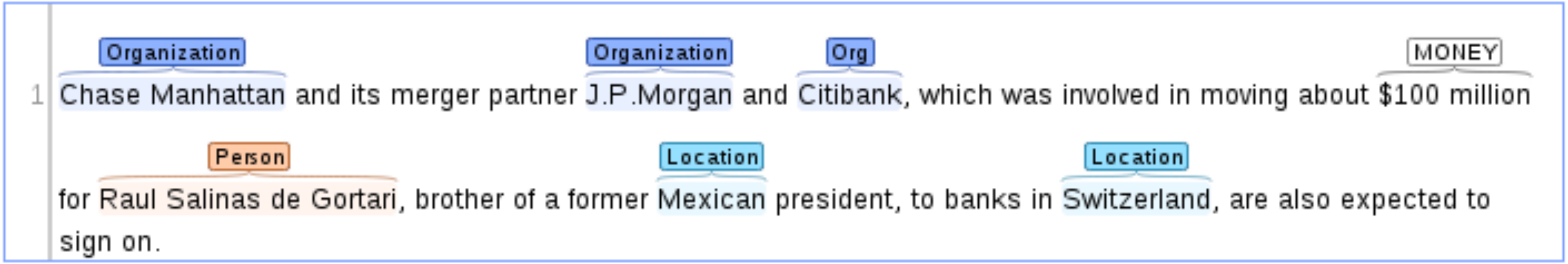
- Useful for higher-level NLP tasks:
  - Chunking
  - Named Entity Recognition
  - Information Extraction
  - Parsing

Stanford CoreNLP Toolkit

Part-of-Speech:



Named Entity Recognition:



Twitter NLP toolkit (Ritter et al.)

Part-of-Speech

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

Named Entity Recognition:

Twitter NLP toolkit (Ritter et al.)

Part-of-Speech

Cant	MD		O
wait	VB		O
for	IN		O
the	DT		O
ravens	NNP	ORG	B-ORG
game	NN		O
tomorrow	NN		O
…	:		O
go	VB		O
ray	NNP	PER	B-PER
rice	NNP		I-PER
!!!!!!!	.		O

Named Entity Recognition as a tagging problem

Tagging (Sequence Labeling)

- Given a sequence (in NLP, words), assign appropriate labels to each word.
- Many NLP problems can be viewed as sequence labeling:
  - POS Tagging
  - Chunking
  - Named Entity Tagging
- Labels of tokens are dependent on the labels of other tokens in the sequence, particularly their neighbors

Plays	well	with	others.
VBZ	RB	IN	NNS

Two Types of Constraints

Influential/JJ members/NNS of/IN the/DT House/NNP Ways/NNP and/CC Means/NNP Committee/NNP introduced/VBD legislation/NN that/WDT would/MD restrict/VB how/WRB the/DT new/JJ savings-and-loan/NN bailout/NN agency/NN can/MD raise/VB capital/NN ./.

- "Local": e.g., *can* is more likely to be a modal verb MD rather than a noun NN
- "Contextual": e.g., a noun is much more likely than a verb to follow a determiner
- Sometimes these preferences are in conflict:

  The trash can is in the garage

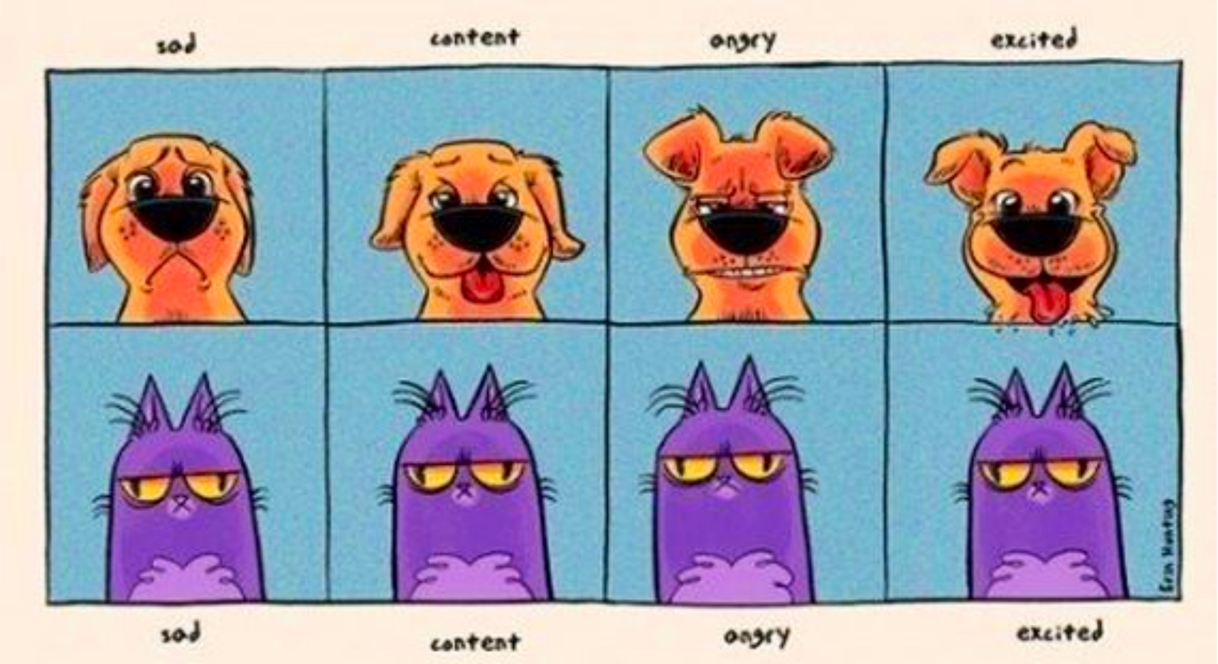
Overview

- The Tagging Problem
- Generative models, and the noisy-channel model, for supervised learning
- Hidden Markov Model (HMM) taggers
  - Basic definitions
  - Parameter estimation
  - The Viterbi algorithm

Supervised Learning Problems

- We have training examples $x^{(i)}, y^{(i)}$ for $i = 1 \ldots m$. Each $x^{(i)}$ is an input, each $y^{(i)}$ is a label.

- Task is to learn a function f mapping inputs x to labels $f(x)$



Supervised Learning Problems

- We have training examples $x^{(i)}, y^{(i)}$ for $i = 1 \ldots m$. Each $x^{(i)}$ is an input, each $y^{(i)}$ is a label.
- Task is to learn a function f mapping inputs x to labels $f(x)$
- Conditional models:
  - Learn a distribution $p(y|x)$ from training examples
  - For any test input x, define $f(x) = \arg\max_y p(y|x)$

Generative Models

- We have training examples $x^{(i)}, y^{(i)}$ for $i = 1 \ldots m$. Task is to learn a function f mapping inputs x to labels $f(x)$.

Generative Models

- We have training examples $x^{(i)}, y^{(i)}$ for $i = 1 \ldots m$. Task is to learn a function f mapping inputs x to labels $f(x)$.

- Generative models:
  - Learn a distribution $p(x, y)$ from training examples
  - Often we have $p(x, y) = p(y)p(x|y)$

Generative Models

- We have training examples $x^{(i)}, y^{(i)}$ for $i = 1 \ldots m$. Task is to learn a function f mapping inputs x to labels $f(x)$.

- Generative models:
  - Learn a distribution $p(x, y)$ from training examples
  - Often we have $p(x, y) = p(y)p(x|y)$

- Note: we then have

$$p(y|x) = \frac{p(y)p(x|y)}{p(x)}$$

where $p(x) = \sum_y p(y)p(x|y)$

Recall the naive Baynes model

Decoding with Generative Models

- We have training examples $x^{(i)}, y^{(i)}$ for $i = 1 \ldots m$. Task is to learn a function f mapping inputs x to labels $f(x)$.

Decoding with Generative Models

- We have training examples $x^{(i)}, y^{(i)}$ for $i = 1 \ldots m$. Task is to learn a function f mapping inputs x to labels $f(x)$.
- Generative models:
  - Learn a distribution $p(x, y)$ from training examples
  - Often we have $p(x, y) = p(y)p(x|y)$

Decoding with Generative Models

- We have training examples $x^{(i)}, y^{(i)}$ for $i = 1 \ldots m$. Task is to learn a function f mapping inputs x to labels $f(x)$.
- Generative models:
  - Learn a distribution $p(x, y)$ from training examples
  - Often we have $p(x, y) = p(y)p(x|y)$
- Output from the model:

$$
\begin{aligned}
f(x) &= \arg\max_y p(y|x) \\
&= \arg\max_y \frac{p(y)p(x|y)}{p(x)} \\
&= \arg\max_y p(y)p(x|y)
\end{aligned}
$$

Recall the naive Baynes model

Overview

- The Tagging Problem
- Generative models, and the noisy-channel model, for supervised learning
- Hidden Markov Model (HMM) taggers
  - Basic definitions
  - Parameter estimation
  - The Viterbi algorithm

Hidden Markov Models

- We have an input sentence $x = x_1, x_2, \ldots, x_n$ (x_i is the i'th word in the sentence)
- We have a tag sequence $y = y_1, y_2, \ldots, y_n$ (y_i is the i'th tag in the sentence)
- We'll use an HMM to define

$$p(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n)$$

  for any sentence $x_1 \ldots x_n$ and tag sequence $y_1 \ldots y_n$ of the same length.
- Then the most likely tag sequence for x is

$$\arg\max_{y_1 \cdots y_n} p(x_1 \ldots x_n, y_1, y_2, \ldots, y_n)$$

Trigram Hidden Markov Models (Trigram HMMs)

For any sentence $x_1 \ldots x_n$ where $x_i \in \mathcal{V}$ for $i = 1 \ldots n$, and any tag sequence $y_1 \ldots y_{n+1}$ where $y_i \in \mathcal{S}$ for $i = 1 \ldots n$, and $y_{n+1} =$ STOP, the joint probability of the sentence and tag sequence is

$$p(x_1 \ldots x_n, y_1 \ldots y_{n+1}) = \prod_{i=1}^{n+1} q(y_i | y_{i-2}, y_{i-1}) \prod_{i=1}^{n} e(x_i | y_i)$$

where we have assumed that $x_0 = x_{-1} = *$.

Parameters of the model:

- $q(s|u, v)$ for any $s \in \mathcal{S} \cup \{\text{STOP}\}$, $u, v \in \mathcal{S} \cup \{*\}$ Trigram parameters
- $e(x|s)$ for any $s \in \mathcal{S}$, $x \in \mathcal{V}$ Emission parameters

An Example

If we have $n = 3$, $x_1 \ldots x_3$ equal to the sentence *the dog laughs*, and $y_1 \ldots y_4$ equal to the tag sequence $\texttt{D N V STOP}$, then

$$
\begin{aligned}
& p(x_1 \ldots x_n, y_1 \ldots y_{n+1}) \\
= \;& q(\texttt{D}|*,*) \times q(\texttt{N}|*,\texttt{D}) \times q(\texttt{V}|\texttt{D},\texttt{N}) \times q(\texttt{STOP}|\texttt{N},\texttt{V}) \\
& \times e(\textit{the}|\texttt{D}) \times e(\textit{dog}|\texttt{N}) \times e(\textit{laughs}|\texttt{V})
\end{aligned}
$$

- STOP is a special tag that terminates the sequence
- We take $y_0 = y_{-1} = *$, where * is a special "padding" symbol