Sequence Labeling

CS 690N, Spring 2017

Advanced Natural Language Processing

http://people.cs.umass.edu/~brenocon/anlp2017/

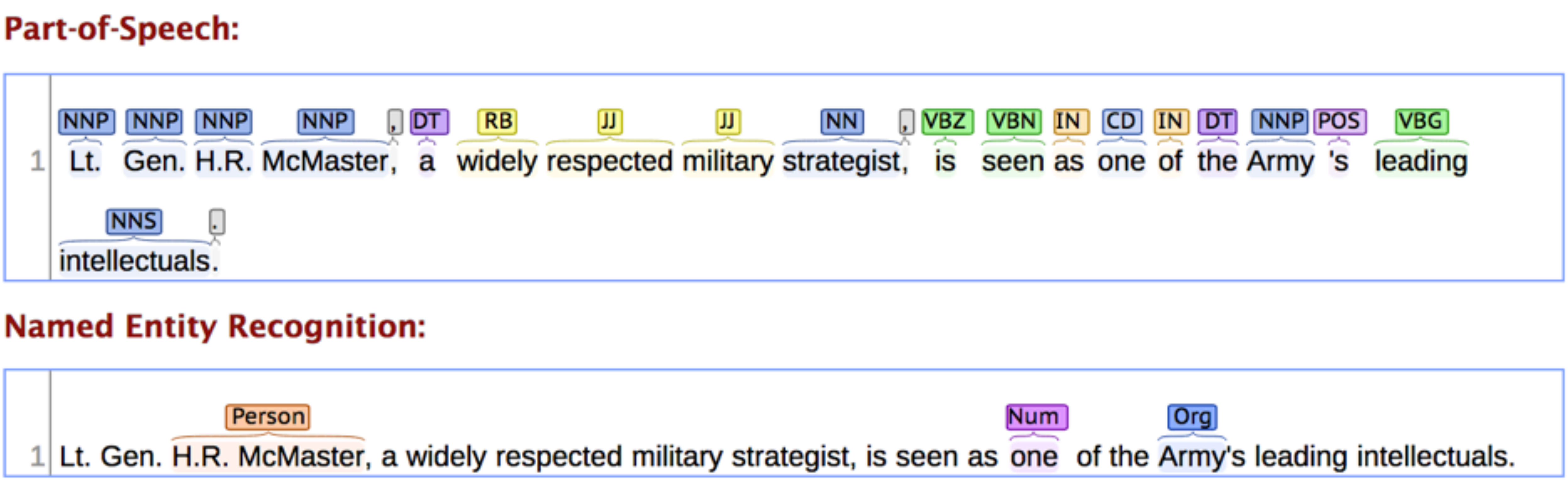
Brendan O'Connor

College of Information and Computer Sciences

University of Massachusetts Amherst

- Sequence labeling problems
  - Part of speech tags
- Models: HMMs, CRFs

- Sequence labeling: from $x_1..x_n$, predict tags $y_1..y_n$
- Named entity recognition:
  an example of *span recognition*
  - BIO tags allow treatment as a sequence labeling problem

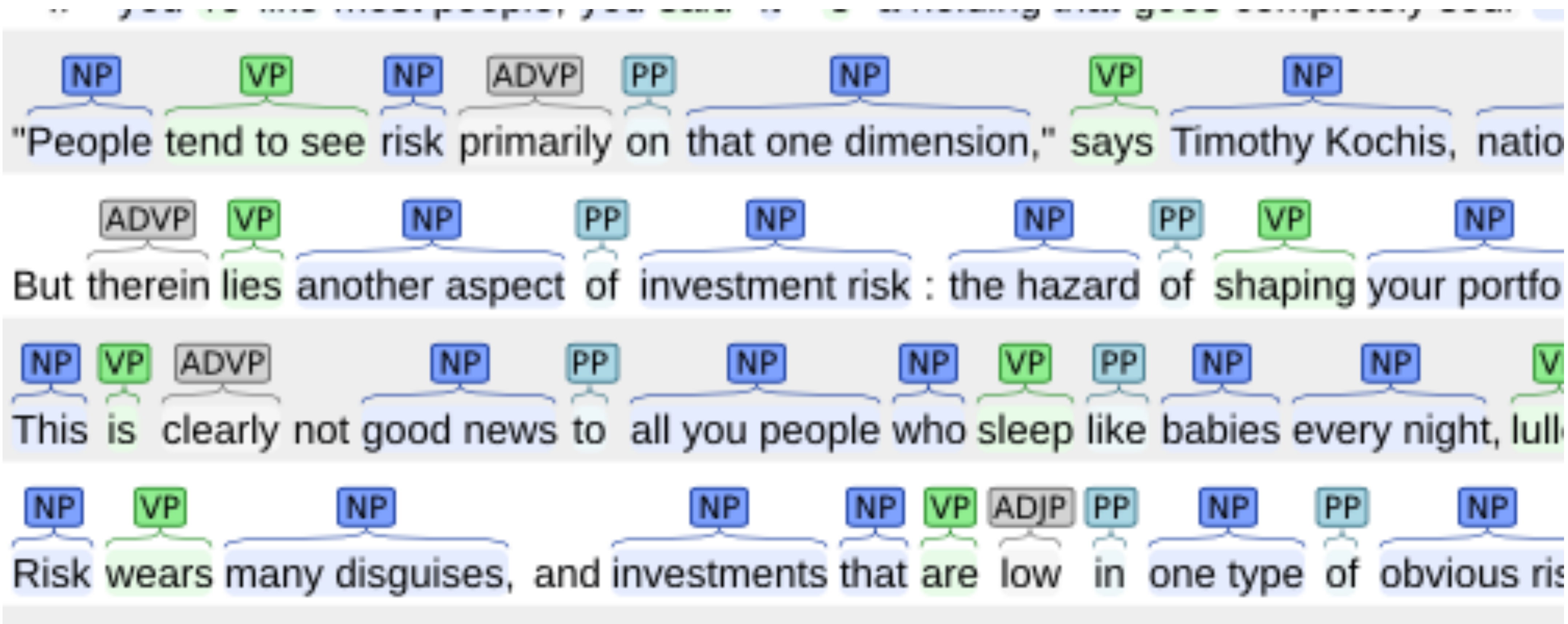


O O B-PER I-PER O O O O . . . .

http://nlp.stanford.edu:8080/corenlp/process

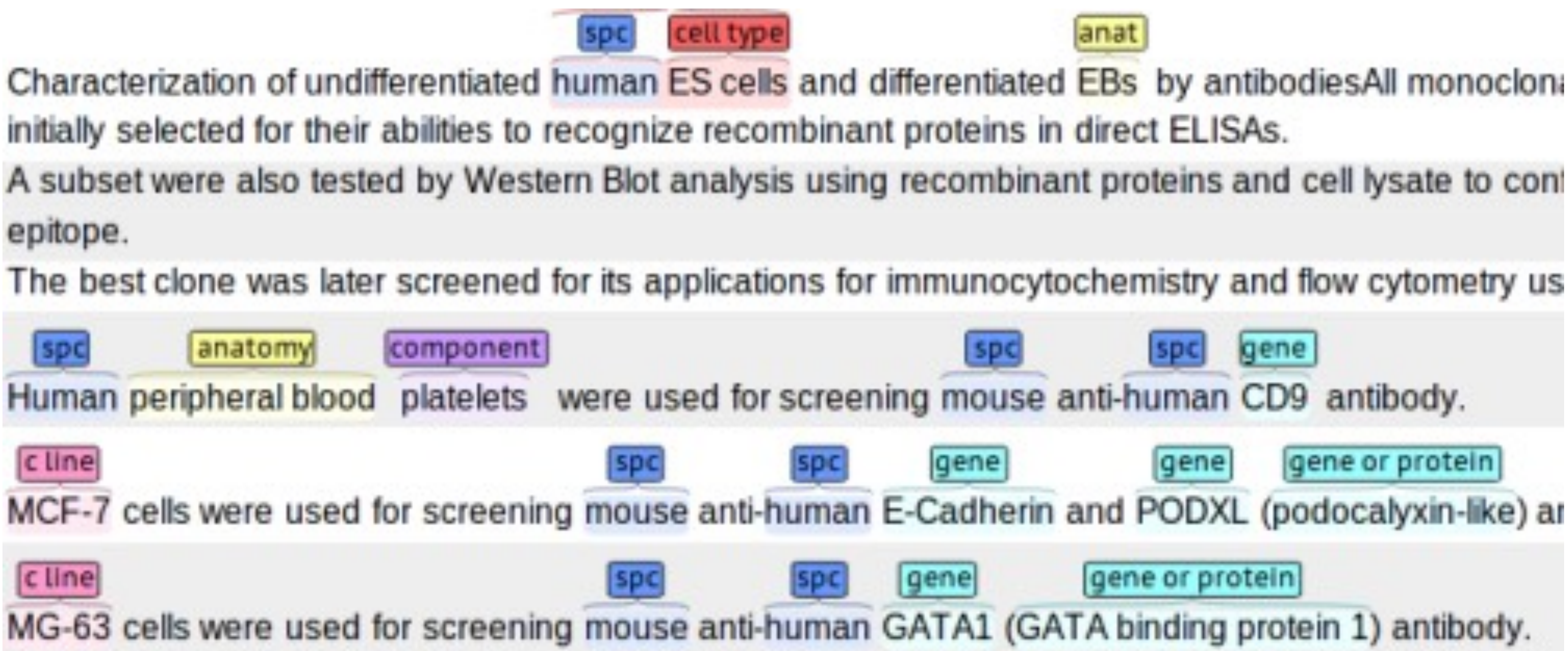
More span labeling tasks

- Syntactic chunking



http://brat.nlplab.org/examples.html

- Biological entities



http://brat.nlplab.org/examples.html

What's a part-of-speech (POS)?

- Syntax = how words compose to form larger meaning-bearing units
- POS = syntactic categories for words
  - You could substitute words within a class and have a syntactically valid sentence.
  - Give information how words can combine.

  - I saw the dog
  - I saw the cat
  - I saw the {table, sky, dream, school, anger, ...}

- (Phrasal/constituent categories generalize this idea. POS tags are constrained to single words.)

Open vs closed classes

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*
- **Modals** *can had*
- **Numbers** *122,312 one*
- **Prepositions** *to with*
- **Particles** *off up*
- **Interjections** *Ow Eh*
- *… more*

slide credit: Chris Manning

ikr	smh	he	asked	fir	yo	last
name	so	he	can	add	u	on
fb	lololol					

ikr	smh	he	asked	fir	yo	last
!	G	O	V	P	D	A
name	so	he	can	add	u	on
N	P	O	V	V	O	P
fb	lololol					
^	!					

Why do we want POS?

- Useful for many syntactic and other NLP tasks.
  - Phrase identification ("chunking")
  - Named entity recognition
  - Full parsing
  - Sentiment
- Especially when there's a low amount of training data

POS patterns: sentiment

- Turney (2002): identify bigram phrases, from unlabeled corpus, useful for sentiment analysis.

Table 1. Patterns of tags for extracting two-word phrases from reviews.

	First Word	Second Word	Third Word (Not Extracted)
1.	JJ	NN or NNS	anything
2.	RB, RBR, or RBS	JJ	not NN nor NNS
3.	JJ	JJ	not NN nor NNS
4.	NN or NNS	JJ	not NN nor NNS
5.	RB, RBR, or RBS	VB, VBD, VBN, or VBG	anything

(plus sentiment PMI stuff)

POS patterns: sentiment

- Turney (2002): identify bigram phrases, from unlabeled corpus, useful for sentiment analysis.

Table 1. Patterns of tags for extracting two-word phrases from reviews.

	First Word	Second Word	Third Word (Not Extracted)
1.	JJ	NN or NNS	anything
2.	RB, RBR, or RBS	JJ	not NN nor NNS
3.	JJ	JJ	not NN nor NNS
4.	NN or NNS	JJ	not NN nor NNS
5.	RB, RBR, or RBS	VB, VBD, VBN, or VBG	anything

Table 2. An example of the processing of a review that the author has classified as *recommended*.[6]

Extracted Phrase	Part-of-Speech Tags	Semantic Orientation
online experience	JJ NN	2.253
low fees	JJ NNS	0.333
local branch	JJ NN	0.421
small part	JJ NN	0.053
online service	JJ NN	2.780
printable version	JJ NN	-0.705
direct deposit	JJ NN	1.288
well other	RB JJ	0.237
inconveniently located	RB VBN	-1.541
other bank	JJ NN	-0.850
true service	JJ NN	-0.732

(plus sentiment PMI stuff)

POS patterns: simple noun phrases

- Quick and dirty noun phrase identification
  http://brenocon.com/JustesonKatz1995.pdf
  http://brenocon.com/handler2016phrases.pdf

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.

POS Tagging: lexical ambiguity

Can we just use a tag dictionary (one tag per word type)?

Types:		WSJ		Brown	
Unambiguous	(1 tag)	44,432	(**86%**)	45,799	(**85%**)
Ambiguous	(2+ tags)	7,025	(**14%**)	8,050	(**15%**)

Most words types are unambiguous ...

- Ambiguous wordtypes tend to be very common ones.
  - I know **that** he is honest = IN (relativizer)
  - Yes, **that** play was nice = DT (determiner)
  - You can't go **that** far = RB (adverb)

POS Tagging: lexical ambiguity

Can we just use a tag dictionary (one tag per word type)?

		WSJ		Brown	
Types:					
Unambiguous	(1 tag)	44,432	(**86%**)	45,799	(**85%**)
Ambiguous	(2+ tags)	7,025	(**14%**)	8,050	(**15%**)
Tokens:					
Unambiguous	(1 tag)	577,421	(**45%**)	384,349	(**33%**)
Ambiguous	(2+ tags)	711,780	(**55%**)	786,646	(**67%**)

Most words types are unambiguous ...

But not so for *tokens*!

- Ambiguous wordtypes tend to be very common ones.
  - I know **that** he is honest = IN (relativizer)
  - Yes, **that** play was nice = DT (determiner)
  - You can't go **that** far = RB (adverb)

Need careful guidelines (and do annotators always follow them?) PTB POS guidelines, Santorini (1990)

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

Some other lexical ambiguities

- Prepositions (P) versus verb particles (T)
  - turn into/P a monster
  - take out/T the trash
  - check it out/T, what's going on/T, shout out/T

Test:
turn slowly into a monster
*take slowly out the trash

Careful annotator guidelines are necessary to define what to do in many cases.

- http://repository.upenn.edu/cgi/viewcontent.cgi?article=1603&context=cis_reports
- http://www.ark.cs.cmu.edu/TweetNLP/annot_guidelines.pdf

Some other lexical ambiguities

- Prepositions (P) versus verb particles (T)
  - turn into/P a monster
  - take out/T the trash
  - check it out/T, what's going on/T, shout out/T

Test:
turn slowly into a monster
*take slowly out the trash

- this,that -- pronouns versus determiners
  - i just orgasmed over this/O
  - this/D wind is serious

Careful annotator guidelines are necessary to define what to do in many cases.

- http://repository.upenn.edu/cgi/viewcontent.cgi?article=1603&context=cis_reports
- http://www.ark.cs.cmu.edu/TweetNLP/annot_guidelines.pdf

How to build a POS tagger?

- Sources of information:
  - POS tags of surrounding words: syntactic context
  - The word itself
  - Features!
    - Word-internal information
    - External lexicons
    - Features from surrounding words

← HMM

← Classifier

← CRF

Sequence labeling

- Seq. labeling as classification
  - Each position *m* gets an independent classification

$$p(y_m \mid w_1..w_n)$$

$$\arg\max_y \theta^\mathsf{T} \mathbf{f}((\mathbf{w}, m), y)$$

$$\begin{aligned}
\boldsymbol{f}((\boldsymbol{w} = \textit{they can fish}, m = 1), \mathrm{N}) =& \langle \textit{they}, \mathrm{N} \rangle \\
\boldsymbol{f}((\boldsymbol{w} = \textit{they can fish}, m = 2), \mathrm{V}) =& \langle \textit{can}, \mathrm{V} \rangle \\
\boldsymbol{f}((\boldsymbol{w} = \textit{they can fish}, m = 3), \mathrm{V}) =& \langle \textit{fish}, \mathrm{V} \rangle .
\end{aligned}$$

Sequence labeling

- Seq. labeling as classification
  - Each position *m* gets an independent classification

$$p(y_m \mid w_1..w_n)$$

$$\arg\max_y \theta^\mathsf{T} \mathbf{f}((\mathbf{w}, m), y)$$

$$\begin{aligned}
\boldsymbol{f}((\boldsymbol{w} = \textit{they can fish}, m = 1), \mathrm{N}) =& \langle \textit{they}, \mathrm{N} \rangle \\
\boldsymbol{f}((\boldsymbol{w} = \textit{they can fish}, m = 2), \mathrm{V}) =& \langle \textit{can}, \mathrm{V} \rangle \\
\boldsymbol{f}((\boldsymbol{w} = \textit{they can fish}, m = 3), \mathrm{V}) =& \langle \textit{fish}, \mathrm{V} \rangle.
\end{aligned}$$

- But syntactic (tag) context is sometimes necessary!
  - *The old man the boat* [garden path sentence]

- Seq. labeling as ***structured prediction***

$$\hat{\boldsymbol{y}}_{1:M} = \underset{\boldsymbol{y}_{1:M} \in \mathcal{Y}(\boldsymbol{w}_{1:M})}{\operatorname{argmax}} \; \boldsymbol{\theta}^{\top} \boldsymbol{f}(\boldsymbol{w}_{1:M}, \boldsymbol{y}_{1:M}),$$

- **Hidden Markov model**
  - Fully generative, simple sequence model
- Supports many operations
  - P(w): Likelihood (generative model)
    - Forward algorithm
  - P(y | w): Predicted sequence ("decoding")
    - Viterbi algorithm
  - P(y_m | w): Predicted tag marginals
    - Forward-Backward algorithm

- The HMM is a type of log-linear model

Forward algorithm

- on board
- stopped here 2/21

Viterbi algorithm

- If the feature function decomposes into local features, dynamic programming gives global solution

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y}}{\operatorname{argmax}}\, \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, \boldsymbol{y}) \qquad\qquad \boldsymbol{f}(\boldsymbol{w}, \boldsymbol{y}) = \sum_{m=1}^{M} \boldsymbol{f}(\boldsymbol{w}, y_m, y_{m-1}, m).$$

- Decompose:

$$\max_{\boldsymbol{y}} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, \boldsymbol{y}) = \max_{\boldsymbol{y}_{1:M}} \sum_{m=1}^{M} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, y_m, y_{m-1}, m)$$

- Define Viterbi variables:

$$v_m(k) \triangleq \max_{\boldsymbol{y}_{1:m-1}} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, k, y_{m-1}, m) + \sum_{n=1}^{m-1} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, y_n, y_{n-1}, n)$$

Viterbi algorithm

- If the feature function decomposes into local features, dynamic programming gives global solution

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y}}{\operatorname{argmax}}\, \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, \boldsymbol{y}) \qquad\qquad \boldsymbol{f}(\boldsymbol{w}, \boldsymbol{y}) = \sum_{m=1}^{M} \boldsymbol{f}(\boldsymbol{w}, y_m, y_{m-1}, m).$$

- Decompose:

$$\max_{\boldsymbol{y}} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, \boldsymbol{y}) = \max_{\boldsymbol{y}_{1:M}} \sum_{m=1}^{M} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, y_m, y_{m-1}, m)$$

$$= \max_{\boldsymbol{y}_{1:M}} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, y_M, y_{M-1}, M) + \sum_{m=1}^{M-1} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, y_m, y_{m-1}, m)$$

- Define Viterbi variables:

$$v_m(k) \triangleq \max_{\boldsymbol{y}_{1:m-1}} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, k, y_{m-1}, m) + \sum_{n=1}^{m-1} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, y_n, y_{n-1}, n)$$

Viterbi algorithm

- If the feature function decomposes into local features, dynamic programming gives global solution

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y}} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, \boldsymbol{y}) \qquad \boldsymbol{f}(\boldsymbol{w}, \boldsymbol{y}) = \sum_{m=1}^{M} \boldsymbol{f}(\boldsymbol{w}, y_m, y_{m-1}, m).$$

- Decompose:

$$\max_{\boldsymbol{y}} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, \boldsymbol{y}) = \max_{\boldsymbol{y}_{1:M}} \sum_{m=1}^{M} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, y_m, y_{m-1}, m)$$

$$= \max_{\boldsymbol{y}_{1:M}} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, y_M, y_{M-1}, M) + \sum_{m=1}^{M-1} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, y_m, y_{m-1}, m)$$

$$= \max_{y_M} \max_{y_{M-1}} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, y_M, y_{M-1}, M) + \max_{\boldsymbol{y}_{1:M-2}} \sum_{m=1}^{M-1} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, y_m, y_{m-1}, m).$$

- Define Viterbi variables:

$$v_m(k) \triangleq \max_{\boldsymbol{y}_{1:m-1}} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, k, y_{m-1}, m) + \sum_{n=1}^{m-1} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, y_n, y_{n-1}, n)$$