Week 4:

POS tags and tagging

School of Information Studies
Syracuse University

Overview

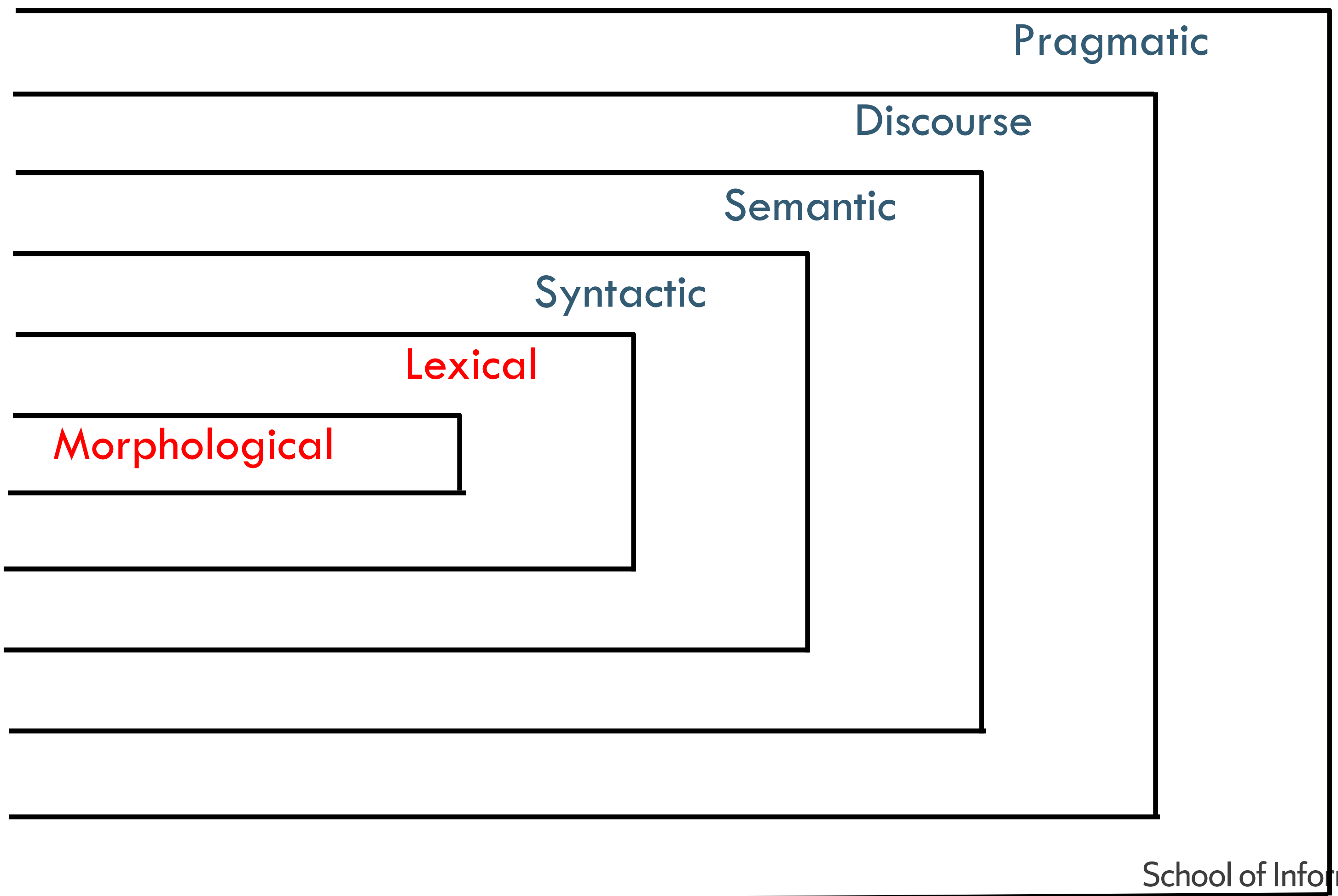
- Mini-talks
- Lecture:
  - POS tags
  - Tagging
- Lab

Part-of-Speech (POS) Tagging: Intro

School of Information Studies
Syracuse University

Synchronic Model of Language

POS tags are assigned to words, but may be determined by adjacent words

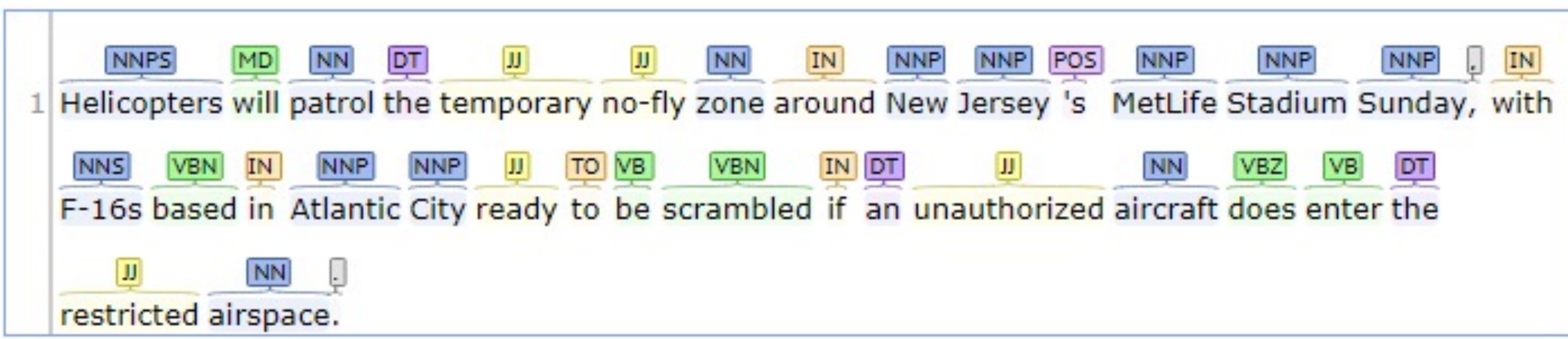


What is Part-Of-Speech Tagging?

The general purpose of a part-of-speech tagger is to associate each word in a text with its correct lexical-syntactic category (represented by a tag)

- Example using tags from the Penn Treebank POS tag set

"Helicopters will patrol the temporary no-fly zone around New Jersey's MetLife Stadium Sunday, with F-16s based in Atlantic City ready to be scrambled if an unauthorized aircraft does enter the restricted airspace."



What are Parts-of-Speech?

- Approximately 8 traditional basic word classes, sometimes called syntactic classes or types

These are the ones taught in grade school grammar

N	noun	*chair, bandwidth, pacing*
V	verb	*study, debate, munch*
ADJ	adjective	*purple, tall, ridiculous (includes articles)*
ADV	adverb	*unfortunately, slowly*
P	preposition	*of, by, to*
CON	conjunction	*and, but*
PRO	pronoun	*I, me, mine*
INT	interjection	*um*

For example, see the shows from "Schoolhouse Rock" on grammar

POS Tag Sets

School of Information Studies
Syracuse University

Possible Tag Sets for English

- Kucera & Francis (Brown Corpus) – 87 POS tags
- C5 (British National Corpus) – 61 POS tags
  - Tagged by Lancaster's UCREL project
- Penn Treebank – 45 POS tags
  - Most widely used of the tag sets today

Penn Treebank

- A corpus containing:
  - over 1.6 million words of hand-parsed material from the Dow Jones News Service, plus an additional 1 million words tagged for part-of-speech.

- Separate licensing needed for commercial use

Word Classes: Penn Treebank Tag Set

Tag	Description	Example	Tag	Description	Example
CC	coordinating 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/subordinating conjunction	*of, in, by*	VBG	verb, gerund	*eating*
JJ	adjective	*yellow*	VBN	verb, past participle	*eaten*
JJR	adjective, comparative	*bigger*	VBP	verb, non-3sg present	*eat*
JJS	adjective, superlative	*wildest*	VBZ	verb, 3sg present	*eats*
LS	list item marker	*1, 2, One*	WDT	wh- determiner	*which, that*
MD	modal	*can, should*	WP	wh- pronoun	*what, who*
NN	noun, singular or mass	*llamas*	WP$	possessive wh-	*whose*

Word Classes: Penn Treebank Tag Set

Tag	Description	Example	Tag	Description	Example
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	*fastest*	.	sentence – final punctuation	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punctuation	: ; ... - -
RP	particle	*up, off*			

Examples of Penn Treebank Tagging

The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.

Book/VB that/DT flight/NN ./.

Does/VBZ that/DT flight/NN serve/VB dinner/NN ?/.

POS Tagging: Introduction

School of Information Studies
Syracuse University

Why is Part-Of-Speech Tagging Needed?

- Knowledge on the function of the word
- Support for the higher levels of NL processing:
  - Phrase Bracketing (use regex with POS tag matching)
    - *E.g., (Det) Adj * N +*
  - Parsing
  - Semantics
- Applications that use POS tagging
  - Speech synthesis - Text-to-speech (how do we pronounce “lead”?)
  - Word-sense disambiguation
  - Sentiment detection — selection of high-opinion or emotion words

Why is Part-Of-Speech Tagging Hard?

- The POS tagging task is to assign a sequence of tags to a sequence of words (usually a sentence)
  - Or can be viewed as assigning a tag to a word in the context of a sequence
- Words may be ambiguous in different ways:
  - A word may have multiple meanings as the same part- of-speech
    - *file* – **noun**, a folder for storing papers
    - *file* – **noun**, instrument for smoothing rough edges
  - A word may function as multiple parts-of-speech
    - a *round* table: **adjective**
    - a *round* of applause: **noun**
    - to *round* out your interests: **verb**
    - to work the year *round*: **adverb**

Overview of Approaches

- **Rule-based Approach**
  - Simple and doesn't require a tagged corpus, but not as accurate as other approaches.

- **Stochastic Approaches**
  - Refers to any approach which incorporates frequencies or probabilities
  - Requires a tagged corpus to learn frequencies of words with POS tags

  - **N-gram taggers**: uses the context of (a few) previous tags

  - **Hidden Markov Model (HMM) taggers**: uses the context of the entire sequence of words and previous tags
    - This technique has been the most widely used of modern taggers, but has the problem of unknown words

Uses the structure of words and parts of words, such as stems

- **Classification Taggers**
  - Uses morphology of word and (a few) surrounding words
  - Helps solve the problem of unknown words

N-gram Approach

N-gram taggers: uses the context of (a few) previous tags

- N-gram approach to probabilistic POS tagging:
  - calculates the probability of a given sequence of tags occurring for a sequence of words
  - the best tag for a given word is determined by the (already calculated) probability that it occurs with the n previous tags
  - may be bi-gram, tri-gram, etc
  - In practice, bigram and trigram probabilities have the problem that the combinations of words are sparse in the corpus

POS Tagging: HMM probabilities

School of Information Studies
Syracuse University

HMM taggers

Hidden Markov Model (HMM) taggers: uses the context of the entire sequence of words and previous tags

- A more comprehensive approach to tagging considers the entire sequence of words
  - *Bolt is expected to race tomorrow*
- What is the best sequence of tags which corresponds to this sequence of observations?
- Probabilistic view:
  - Consider all possible sequences of tags
  - Out of this universe of sequences, choose the tag sequence which is most probable given the observation sequence of n words: $W_1 \ldots W_n$.

HMM decoding

- We want, out of all sequences of n tags $t_1 \ldots t_n$ the single tag sequence such that $P(t_1 \ldots t_n | w_1 \ldots w_n)$ is highest.
  - i.e. the probability of the tag sequence $t_1 \ldots t_n$ given the word sequence $w_1 \ldots w_n$

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

Hat ^ means "our estimate of the best one"

Argmax$_x$ f(x) means "the x such that f(x) is maximized"

- i.e. find the tag sequence that maximizes the probability

Road to HMMs

This equation is guaranteed to give us the best tag sequence

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

But how to make it operational? How to compute this value?

Intuition of Bayesian classification:

- Use Bayes rule to transform into a set of other probabilities that are easier to compute

Thomas Bayes 1701 - 1761

Using Bayes Rule

Bayes rule:

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

Apply Bayes Rule:

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

Note that this is **<u>using the conditional probability</u>**, <u>given a tag sequence, what is the most likely word sequence with those tags</u>.

- Drop denominator $P(w_1^n)$ as it is the same for every sequence

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

Likelihood and *Prior*

- Further simplify

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

- **Likelihood**: <u>assume that the probability of the word depends only on its tag</u>

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

- **Prior**: <u>use the bigram assumption that the tag only depends on the previous tag</u>

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

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

Two Sets of Probabilities (1)

1. Tag transition probabilities $p(t_i|t_{i-1})$ (priors)

- Determiners(DT) likely to precede adjs(JJ) and nouns(NN)
  - That/DT flight/NN
  - The/DT yellow/JJ hat/NN
  - So we expect P(NN|DT) and P(JJ|DT) to be high

- Compute P(NN|DT) by counting in a labeled corpus:

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

Count of DT NN sequence

$$P(NN|DT) = \frac{C(DT, NN)}{C(DT)} = \frac{56,509}{116,454} = .49$$

Two Sets of Probabilities (2)

2. Word likelihood probabilities $p(w_i|t_i)$

- VBZ (3sg Pres verb) likely to be "is"
- Compute P(is|VBZ) by counting in a labeled corpus:

$$P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$$

Count of "is" tagged with VBZ

$$P(is|VBZ) = \frac{C(VBZ, is)}{C(VBZ)} = \frac{10,073}{21,627} = .47$$

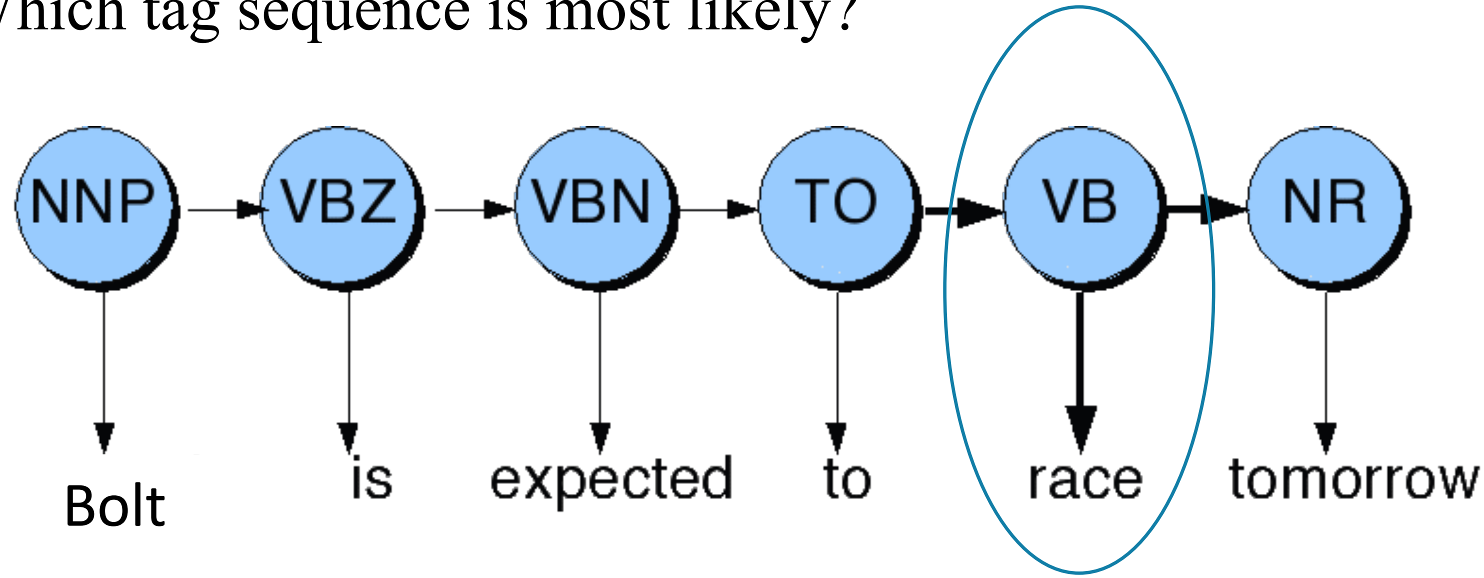
An Example: the word “race”

- The word “race” can occur as a verb or as a noun:
  - Bolt/NNP is/VBZ expected/VBN to/TO **race**/VB tomorrow/NR
  - People/NNS continue/VB to/TO inquire/VB the/DT reason/NN for/IN the/DT **race**/NN for/IN outer/JJ space/NN
- How do we pick the right tag?

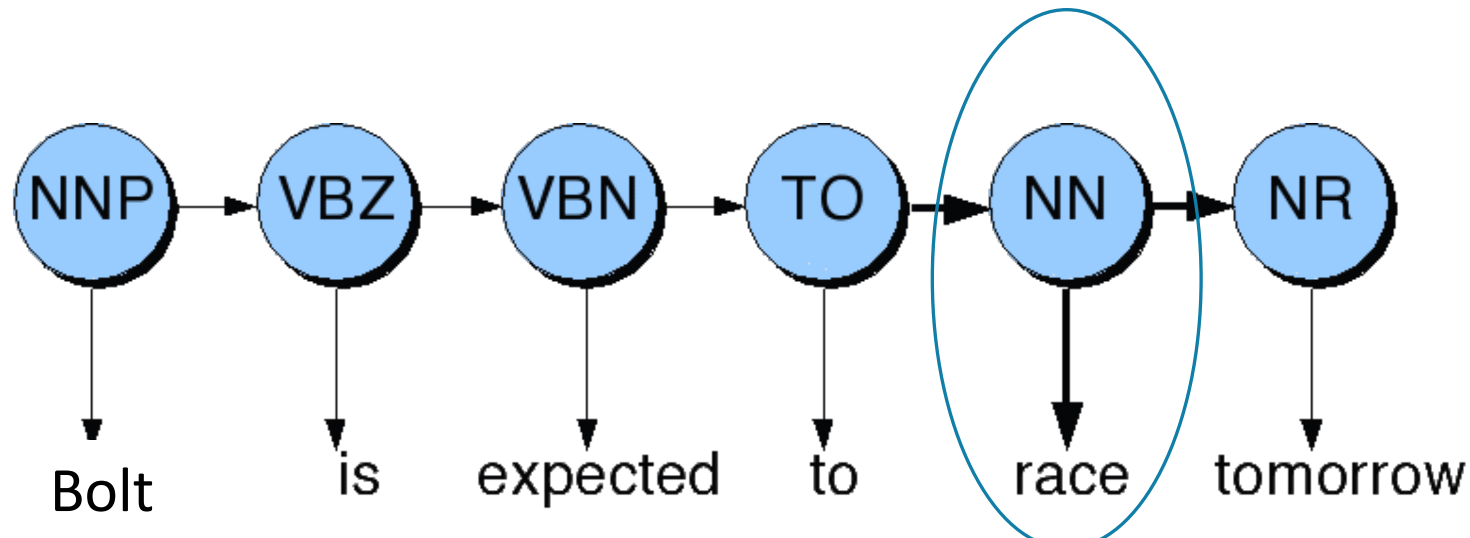
Disambiguating "race"

Which tag sequence is most likely?

(a)



(b)



Example

The equations only differ in "to race tomorrow"

$P(NN|TO) = .00047$

$P(VB|TO) = .83$

The tag transition probabilities P(NN|TO) and P(VB|TO)

$P(race|NN) = .00057$

$P(race|VB) = .00012$

Lexical likelihoods from the Brown corpus for 'race' given a POS tag NN or VB.

$P(NR|NN) = .0012$

$P(NR|VB) = .0027$

Tag sequence probability for the likelihood of an adverb occurring given the previous tag verb or noun

$P(VB|TO)P(NR|VB)P(race|VB) = .00000027$

$P(NN|TO)P(NR|NN)P(race|NN)=.00000000032$

So we (correctly) choose the verb tag.

In-class activity

- Calculate the probabilities for the POS tags (IN/ RB) of word "around" in this sentence :

  Meet/VB me/PRP around/(IN or RB) the/DT corner/NN.

Here are a list of probabilities that you might need:

- P(around | IN)=0.03
- P(IN | PRP)= 0.007
- P(DT | IN)=0. 025
- P(around | RB)= 0.04
- P(RB | PRP)=0.0001
- P(DT | RB)=0.009

Post your calculation of the probabilities and the selected tag in **Discussion**.

POS Tagging: Classifier

School of Information Studies
Syracuse University

Feature-based Classifiers

- A feature-based classifier is an algorithm that will take a word and assign a POS tag based on **features of the word** in its context in the sentence.
  - typically feature classifier uses information from 1-3 surrounding words on either side

    Word_{-2} Word_{-1} Word Word_{+1} Word_{+2}

- Many algorithms are used for these traditional classifiers, including
  - Naïve Bayes
  - Maximum Entropy (MaxEnt)
  - Support Vector Machines (SVM)

Features of words

- We can do surprisingly well just looking at a word by itself:
  - Word — the: the → DT (determiner)
  - Prefixes — unhappy: un- → JJ (adjective)
  - Suffixes — Importantly: -ly → RB
tangential: -al → JJ
  - Capitalization — Syracuse: CAP → NNP (proper noun)
  - Word shapes — 35-year: d-x → JJ
- These properties can include information about the previous or the next word(s)
  - The word *"be"* appears to the left of "pretty" → JJ
- But not information about **tags** of the previous or next words, unlike HMM

Development process for features

- The tagged data should be separated into a training set and a test set.
  - The classifier is trained on the training set, which produces a "tagger"
  - And evaluated on the test set, by applying the tagger to every word and comparing the predicted tag with the answer in the test set

- If our feature-based tagger has errors, then we improve the features.
  - Suppose we incorrectly tag *as* as IN in the phrase *as soon as*, when it should be RB:

PRP	VBD	IN	RB	IN	PRP	VBD	.
They	left	as	soon	as	he	arrived	.

  - We could fix this with a feature that include the next word.

POS Tagging: Evaluation and Demos

School of Information Studies
Syracuse University

Evaluation: Is our POS tagger any good?

- Answer: we use a manually tagged corpus, which we will call the "Gold Standard"
  - We run our POS tagger on the gold standard and compare its predicted tags with the gold tags
  - We compute the accuracy (and other evaluation measures)
- Important: 100% is impossible even for human annotators.
  - We estimate humans can do POS tagging at about 98% accuracy.
  - Some tagging decisions are very subtle and hard to do:
    - All/DT we/PRP gotta/VBN do/VB is/VBZ go/VB **around/IN** the/DT corner/NN
    - Subaru/NNP Outback/NNP costs/VBZ **around**/RB 25000/CD
  - The "Gold Standard" will have human mistakes; humans are subject to fatigue, etc.

Overview of POS tagger Accuracies

- Stanford NLP group performed experiments with different tagging techniques and looked at the improvements.

Rough accuracies: all words / unknown words

- Most freq tag: ~90% / ~50%
- Trigram HMM: ~95% / ~55%
  - HMM with trigrams
- Maxent P(t|w): 93.7% / 82.6%
  - Feature based tagger
- MEMM tagger: 96.9% / 86.9%
  - Combines feature based and HMM tagger
- Upper bound: ~98% (human agreement)

Most errors on unknown words

* *Results from slides by Chris Manning*

POS taggers with online demos

- Many pages list downloadable taggers (and other resources)
  - http://nlp.stanford.edu/software/tagger.shtml
  - https://cogcomp.seas.upenn.edu/page/software_view/POS

- There are not too many on-line taggers available for demos, but here are some possibilities:
  - The Stanford online parser demo includes POS tags: http://corenlp.run/

Conclusions

- POS tagging is a doable task with high performance results
  - In addition to the standard text POS taggers discussed here, there are now POS tag systems and taggers developed for social media text.
- Contributes to many practical, real-world NLP applications and is now used as a pre-processing module in most systems
- Computational techniques learned at this level can be applied to NLP tasks at higher levels of language processing

Lab

School of Information Studies
Syracuse University

Tasks

1. Demo tagging
   - Brown corpus– its own POS tags
   - Penn treebank POS
2. POS Tagging
3. N-gram tagger
   - unigram tagging
   - bigram tagger
   - bigram tagger with backoff