CS 291A: Deep Learning for NLP

A Crash Course on NLP tasks

William Wang
UCSB Computer Science
william@cs.ucsb.edu

Slides adapted from Jason Eisner.

Announcements

- If you just joined us from the waitlist, or you have not yet chosen your research paper to present:
  - Please see me immediately after the class.

- Also, please bring laptop or phone with Piazza app pre-installed to each class.

NLP Tasks

1. Classify the entire document ("text categorization")

Sentiment classification

What features of the text could help predict # of stars?
(e.g., using a log-linear model) How to identify more?
Are the features hard to learn? (syntax? sarcasm?)

An extremely versatile machine!, November 22, 2006

By Dr. Nickolas E. Jorgensen "njorgens3"

This review is from: Cuisinart DGB-600BC Grind & Brew, Brushed Chrome (Kitchen)

This coffee-maker does so much! It makes weak, watery coffee! It grinds beans if you want it to! It inexplicably floods the entire counter with half-brewed coffee when you aren't looking! Perhaps it could be used to irrigate crops... It is time-consuming to clean, but in fairness I should also point out that the stainless-steel thermal carafe is a durable item that has withstood being hurled onto the floor in rage several times. And if all these features weren't enough, it's pretty expensive too. If faced with the choice between having a car door repeatedly slamming into my genitalia and buying this coffee-maker, I'd unhesitatingly choose the Cuisinart! The coffee would be lousy, but at least I could still have children...

Other text categorization tasks

- Is it spam?
- What medical billing code for this visit?
- What grade, as an answer to this essay question?
- Is it interesting to this user?
  - News filtering; helpdesk routing
  - Detect levels-of-interest (Wang et al., 2013 Comp. Speech& Lang)
- Is it interesting to this NLP program?
  - If it's Spanish, translate it from Spanish
  - If it's subjective, run the sentiment classifier
  - If it's an appointment, run information extraction
- Where should it be filed?
  - Which mail folder? (work, friends, junk, urgent ...)
  - Yahoo! / Open Directory / digital libraries

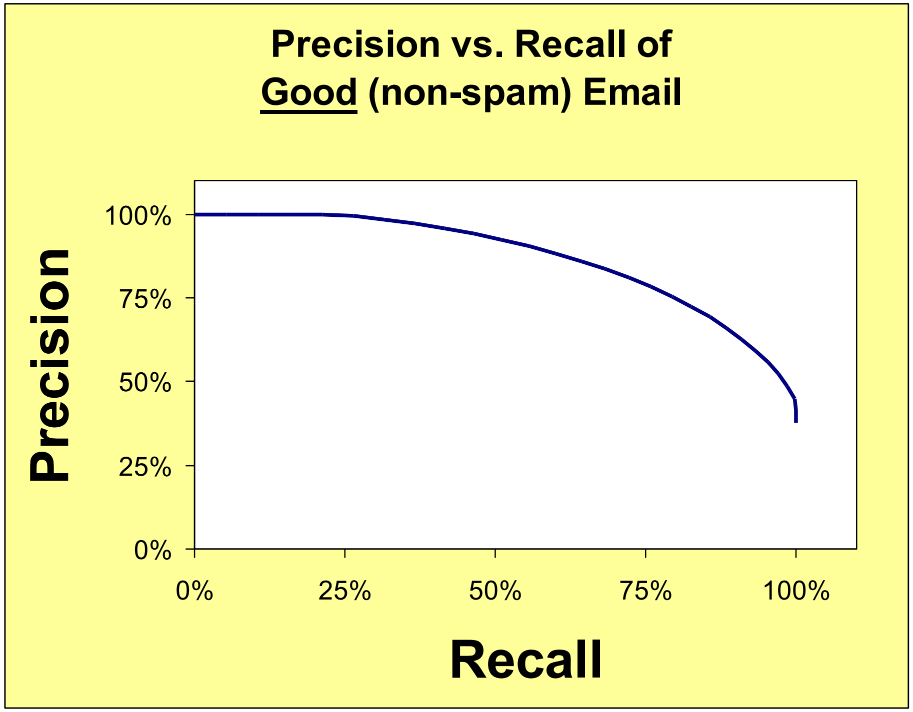
Measuring Performance

- Classification accuracy: What % of messages were classified correctly?
- **Is this what we care about?**

	Overall accuracy	Accuracy on spam	Accuracy on gen
System 1	95%	99.99%	90%
System 2	95%	90%	99.99%

- Which system do you prefer?

Measuring Performance



- Precision =

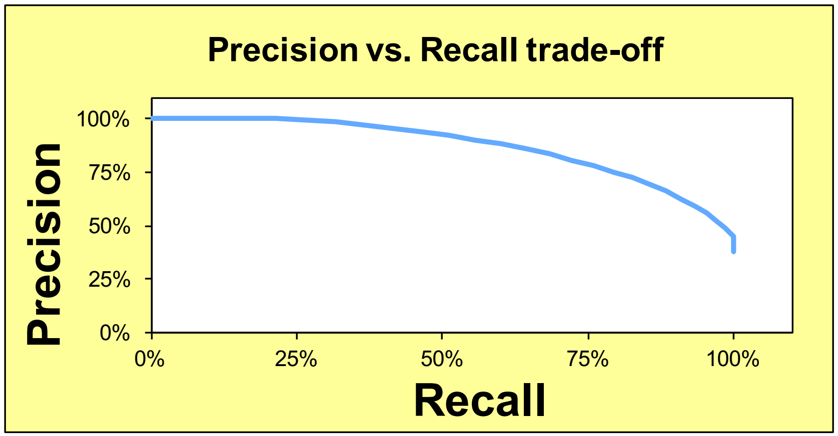
$$\frac{\text{good messages kept}}{\text{all messages kept}}$$

- Recall =

$$\frac{\text{good messages kept}}{\text{all good messages}}$$

Move from high precision to high recall by
deleting fewer messages (delete only if spamminess > high threshold)

Measuring Performance: Search Engine



- Precision =

$$\frac{\text{relevant docs}}{\text{all retrieved docs}}$$

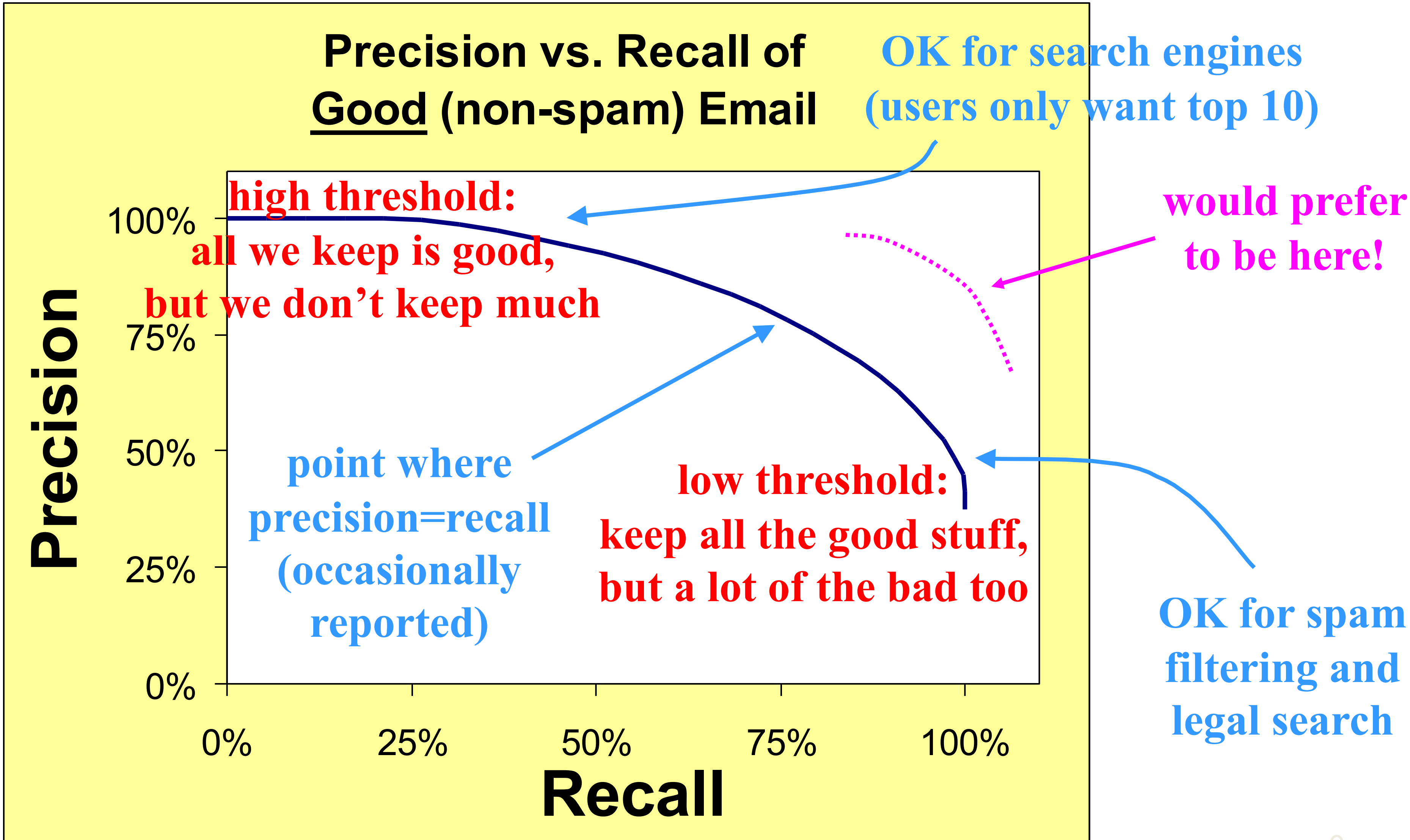
- Recall =

$$\frac{\text{retrieved relevant docs}}{\text{all relevant docs}}$$

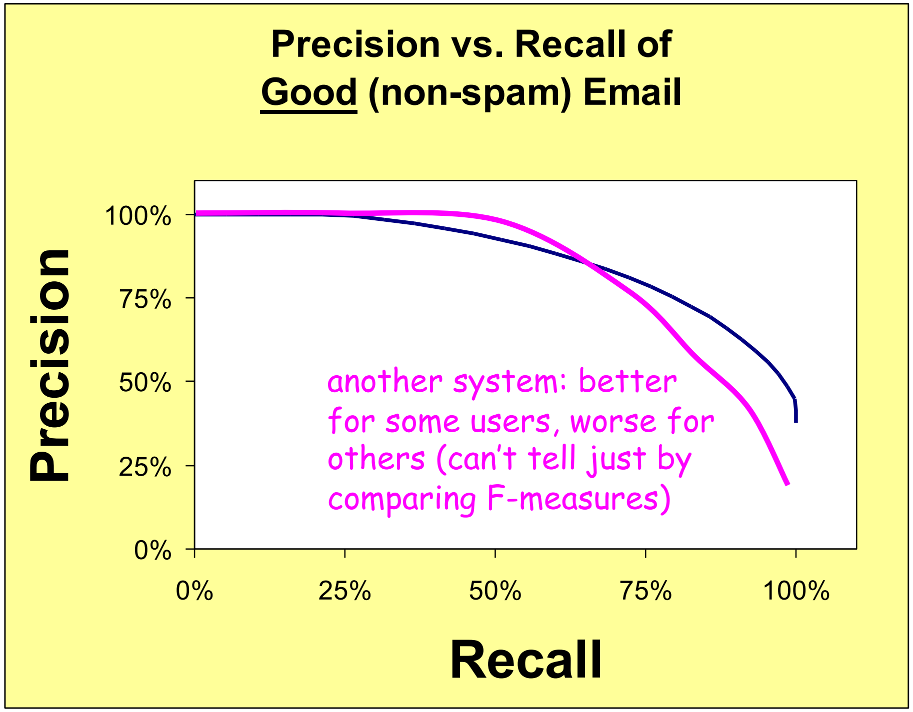
Piazza Poll:

what should search engine optimize?

Measuring Performance



Measuring Performance



- Precision = $\dfrac{\text{good messages kept}}{\text{all messages kept}}$
- Recall = $\dfrac{\text{good messages kept}}{\text{all good messages}}$
- F-measure = $\left(\dfrac{\text{precision}^{-1} + \text{recall}^{-1}}{2}\right)^{-1}$

Move from high precision to high recall by
deleting fewer messages (raise threshold)

Conventional to tune system and threshold to optimize F-measure on dev data

But it's more informative to report the whole curve

Since in real life, the user should be able to pick a tradeoff point they like

Supervised Learning Methods

- Conditional log-linear models are a good hammer
  - Feature engineering: Throw in enough features to fix most errors
  - Training: Learn weights θ such that in training data, the true answer tends to have a **high probability**
  - Test: Output the highest-probability answer

- The most popular alternatives are
  - Perceptron, SVM, neural network, …
  - These also learn a linear/non-linear scoring function
  - Learner just seeks weights θ such that in training data, the desired answer has a **higher score** than the wrong answers

Supervised Learning Methods

- Easy to build a "yes" or "no" predictor from supervised training data
  - Plenty of software packages to do the learning & prediction
  - Lots of people in NLP never go beyond this ☺

- Similarly, easy to build a system that chooses from a small finite set
  - Basically the same deal

- A lot of the recent successes on deep learning are based on supervised learning:
  - Feed-forward neural networks / Multi-layer perceptron's
  - Convolutional neural networks
  - Recurrent neural networks

NLP Tasks

1. Classify the entire document
2. Classify individual word tokens

p(class | token in context)

Word Sense Disambiguation (WSD)

Problem:

The company said the *plant* is still operating ...

⇒ (A) Manufacturing plant or

⇒ (B) Living plant

Why do we care about WSD?
Can you give a related application that matters?

" "	... computer disk drive *plant* located in ...
" "	company manufacturing *plant* is in Orlando ...
(2) Living	... animal rather than *plant* tissues can be ...
" "	... to strain microscopic *plant* life from the ...
" "	and Golgi apparatus of *plant* and animal cells

Test Data:

Sense	Context
???	... vinyl chloride monomer *plant* , which is ...
???	... molecules found in *plant* tissue from the ...

slide courtesy of D. Yarowsky

p(class | token in context)

WSD for Machine Translation

(English → Spanish)

Problem:

... He wrote the last **sentence** two years later ...

⇒ *sentencia* (legal sentence) or

⇒ *frase* (grammatical sentence)

Training Data: Build a special classifier just for tokens of "sentence"

Translation	Context
(1) sentencia	... for a maximum *sentence* for a young offender ...
" "	... of the minimum *sentence* of seven years in jail ...
" "	... were under the *sentence* of death at that time ...
(2) frase	... read the second *sentence* because it is just as ...
" "	... The next *sentence* is a very important ...
" "	... It is the second *sentence* which I think is at ...

Test Data:

Translation	Context
???	... cannot criticize a *sentence* handed down by ...
???	... listen to this *sentence* uttered by a former ...

slide courtesy of D. Yarowsky

p(class | token in context)

Accent Restoration in Spanish & French

Problem:

Input:	... deja travaille cote a cote ...
	⇓
Output:	... déjà travaillé côte à côte ...

Examples:

... appeler l'autre **cote** de l'atlantique ...
⇒ *côté* (meaning side) or
⇒ *côte* (meaning coast)

. . . une famille des **pecheurs** . . .
⇒ *pêcheurs* (meaning fishermen) or
⇒ *pécheurs* (meaning sinners)

slide courtesy of D. Yarowsky

p(class | token in context)

Accent Restoration in Spanish & French

Training Data:

Pattern	Context
(1) côté	... du laisser de *cote* faute de temps ...
" "	... appeler l' autre *cote* de l' atlantique ...
" "	... passe de notre *cote* de la frontiere ...
(2) côte	... vivre sur notre *cote* ouest toujours ...
" "	... creer sur la *cote* du labrador des ...
" "	travaillaient cote a *cote* , ils avaient ...

Test Data:

Pattern	Context
???	... passe de notre *cote* de la frontiere ...
???	... creer sur la *cote* du labrador des ...

p(class | token in context)

Capitalization Restoration

Problem:

… FRIED CHICKEN, **TURKEY** SANDWICHES AND FROZEN ...

⇒ *turkey* (the *bird*) or

⇒ *Turkey* (the *country*)

Training Data:

Capitalization	Context
(1) turkey	… OF FRIED CHICKEN , TURKEY SANDWICHES AND FROZEN …
” ”	… NTS A POUND , WHILE TURKEY PRICES ROSE 1.2 CENTS …
” ”	… PLAY , REAL GRADE-A TURKEY , WHICH ONLY A PRICE …
(2) Turkey	… INUNDATED EASTERN TURKEY AFTER THE EARLIER …
” ”	… FEELINGS TOWARD TURKEY SURFACED WHEN GREECE …
” ”	… THE CONTRACT WITH TURKEY WILL PROVIDE OPPORTU…

Test Data:

Capitalization	Context
???	… NECK LIKE THAT OF A TURKEY ON A CHOPPING BLOCK …
???	… PROBLEM IS THAT TURKEY IS NOT A EUROPEAN …

slide courtesy of D. Yarowsky

p(class | token in context)

Text-to-Speech Synthesis

Problem:

... slightly elevated *lead* levels ...

⇒ *lɛd* (as in *lead mine*) or

⇒ *li:d* (as in *lead role*)

Training Data:

Pronunciation	Context
(1) lɛd	... it monitors the *lead* levels in drinking ...
” ”	... conference on *lead* poisoning in ...
” ”	... strontium and *lead* isotope zonation ...
(2) li:d	... maintained their *lead* Thursday over ...
” ”	... to Boston and *lead* singer for Purple ...
” ”	... Bush a 17-point *lead* in Texas , only 3 ...

Test Data:

Pronunciation	Context
???	... median blood *lead* concentration was ..
???	... his double-digit *lead* nationwide . The ...

slide courtesy of D. Yarowsky

p(class | token in context)

Spelling Correction

Problem:

... and he fired presidential **aid/aide** Dick Morris after ...

⇒ *aid* or

⇒ *aide*

Training Data:

Spelling	Context
(1) aid	... and cut the foreign *aid/aide* budget in fiscal 1996 ...
" "	... they offered federal *aid/aide* for flood-ravaged states ...
(2) aide	... fired presidential *aid/aide* Dick Morris after ...
" "	... and said the chief *aid/aide* to Sen. Baker, Mr. John ...

Test Data:

Spelling	Context
???	... said the longtime *aid/aide* to the Mayor of St. ...
???	... will squander the *aid/aide* it receives from the ...

slide courtesy of D. Yarowsky

What features? Example: "word to left"

Word to left	Frequency as **Aid**	Frequency as **Aide**
foreign	718	1
federal	297	0
western	146	0
provide	88	0
covert	26	0
oppose	13	0
future	9	0
similar	6	0
presidential	0	63
chief	0	40
longtime	0	26
aids-infected	0	2
sleepy	0	1
disaffected	0	1
indispensable	2	1
practical	2	0
squander	1	0

Spelling correction using an n-gram language model ($n \geq 2$) would use words to left and right to help predict the true word.

Similarly, an HMM would predict a word's class using classes to left and right.

But we'd like to throw in all kinds of other features, too …

An assortment of possible cues ...

	Position	Collocation	lɛd	li:d
N-grams	+1 L	lead *level/N*	219	0
	-1 W	*narrow* lead	0	70
(word,	+1 W	lead *in*	207	898
lemma,	-1W,+1W	*of* lead *in*	162	0
part-of-speech)	-1W,+1W	*the* lead *in*	0	301
	+1P,+2P	lead , <*NOUN*>	234	7
Wide-context	$\pm$k W	*zinc* (in $\pm k$ words)	235	0
collocations	$\pm$k W	*copper* (in $\pm k$ words)	130	0
Verb-object	-V L	*follow/V* + lead	0	527
relationships	-V L	*take/V* + lead	1	665

generates a whole bunch of potential cues – use data to find out which ones work best

Word to left	Frequency as **Aid**	Frequency as **Aide**
foreign	718	1
federal	297	0
western	146	0
provide	88	0

An assortment of possible cues ...

	Position	Collocation	lɛd	li:d
N-grams	+1 L	lead *level/N*	219	0
	-1 W	*narrow* lead	0	70
(word,	+1 W	lead *in*	207	898
lemma,	-1W,+1W	*of* lead *in*	162	0
part-of-speech)	-1W,+1W	*the* lead *in*	0	301
	+1P,+2P	lead , <*NOUN*>	234	7
Wide-context	$\pm$k W	*zinc* (in $\pm k$ words)	235	0
collocations	$\pm$k W	*copper* (in $\pm k$ words)	130	0
Verb-object	-V L	*follow/V* + lead	0	527
relationships	-V L	*take/V* + lead	1	665

This feature is relatively weak, but weak features are still useful, especially since very few features will fire in a given context.

11.40	*follow/V* + lead	$\Rightarrow$ li:d
11.20	*zinc* (in $\pm k$ words)	$\Rightarrow$ lɛd
11.10	lead *level/N*	$\Rightarrow$ lɛd
10.66	*of* lead *in*	$\Rightarrow$ lɛd
10.59	*the* lead *in*	$\Rightarrow$ li:d
10.51	lead *role*	$\Rightarrow$ li:d

merged ranking
of all cues
of all these types

Final decision list for ***lead*** (abbreviated)

List of all features, ranked by their weight.

(These weights are for a simple "decision list" model where the single highest-weighted feature that fires gets to make the decision all by itself.)

LogL	Evidence	Pronunciation
11.40	*follow/V* + lead	⇒ li:d
11.20	*zinc* (in $\pm k$ words)	⇒ lɛd
11.10	lead *level/N*	⇒ lɛd
10.66	*of* lead *in*	⇒ lɛd
10.59	*the* lead *in*	⇒ li:d
10.51	lead *role*	⇒ li:d
10.35	*copper* (in $\pm k$ words)	⇒ lɛd
10.28	lead *time*	⇒ li:d
10.24	lead *levels*	⇒ lɛd
10.16	lead *poisoning*	⇒ lɛd
8.55	*big* lead	⇒ li:d
8.49	*narrow* lead	⇒ li:d
7.76	*take/V* + lead	⇒ li:d
5.99	lead , *NOUN*	⇒ lɛd
$\boxed{1.15}$	lead *in*	⇒ li:d
	○ ○ ○	

Piazza Poll:

How would you combine all evidence?

LogL	Evidence	Pronunciation
11.40	*follow/V* + lead	⇒ li:d
11.20	*zinc* (in $\pm k$ words)	⇒ lɛd
11.10	lead *level/N*	⇒ lɛd
10.66	*of* lead *in*	⇒ lɛd
10.59	*the* lead *in*	⇒ li:d
10.51	lead *role*	⇒ li:d
10.35	*copper* (in $\pm k$ words)	⇒ lɛd
10.28	lead *time*	⇒ li:d
10.24	lead *levels*	⇒ lɛd
10.16	lead *poisoning*	⇒ lɛd
8.55	*big* lead	⇒ li:d
8.49	*narrow* lead	⇒ li:d
7.76	*take/V* + lead	⇒ li:d
5.99	lead , *NOUN*	⇒ lɛd
1.15	lead *in*	⇒ li:d
	○○○	

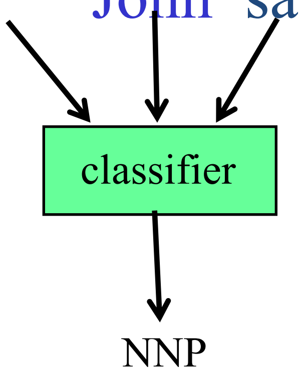
Part of Speech Tagging

- We could treat tagging as a token classification problem
  - Tag each word independently given features of context
  - And features of the word's spelling (suffixes, capitalization)

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

John saw the saw and decided to take it to the table.

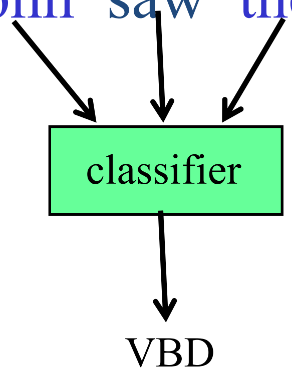


Slide from Ray Mooney

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

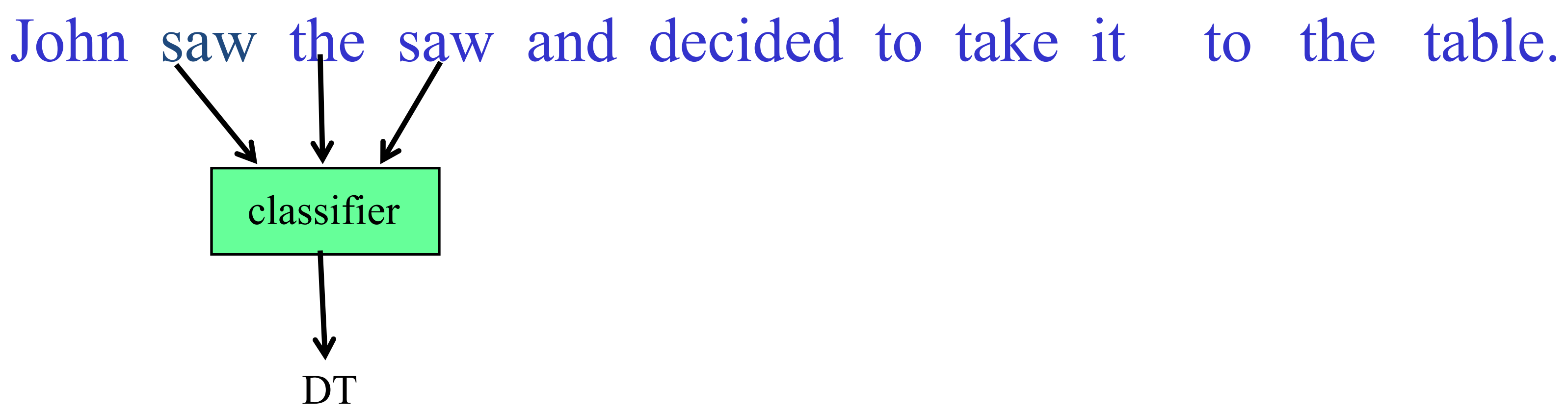
John saw the saw and decided to take it to the table.



Slide from Ray Mooney

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

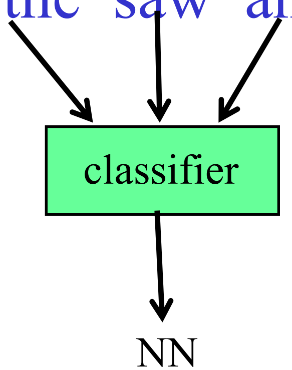


Slide from Ray Mooney

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

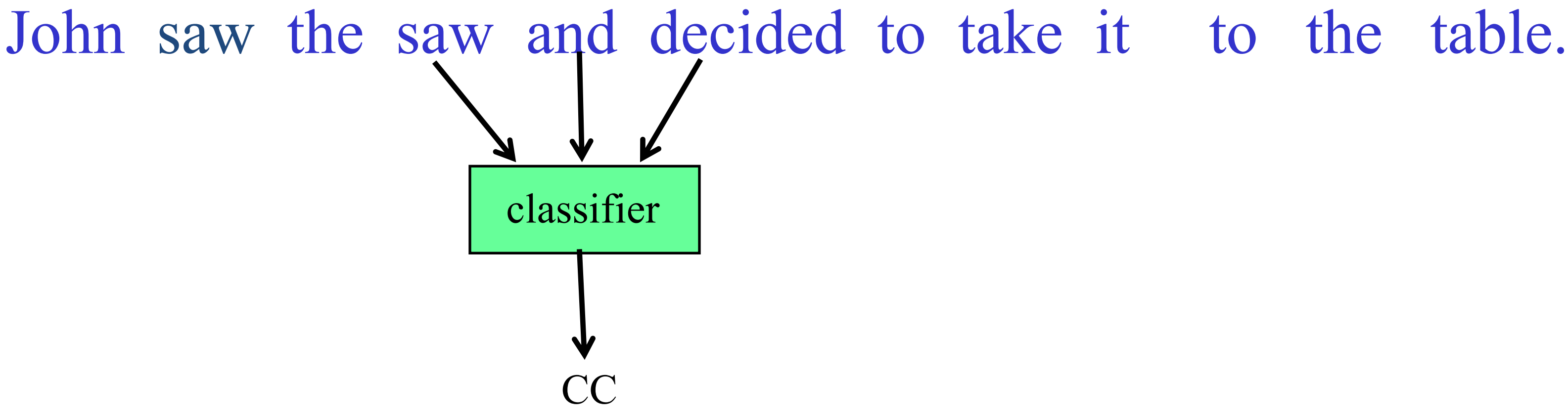
John saw the saw and decided to take it to the table.



Slide from Ray Mooney

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

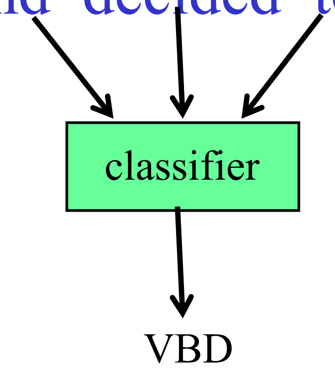


Slide from Ray Mooney

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

John saw the saw and decided to take it to the table.

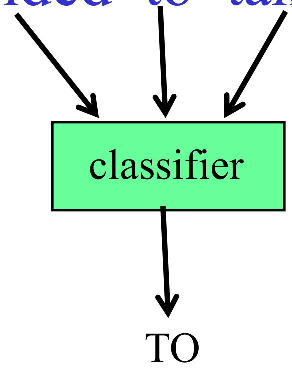


Slide from Ray Mooney

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

John saw the saw and decided to take it to the table.

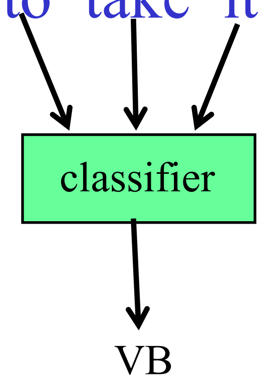


Slide from Ray Mooney

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

John saw the saw and decided to take it to the table.

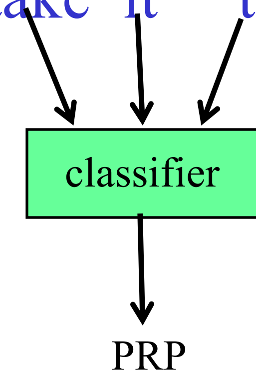


Slide from Ray Mooney

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

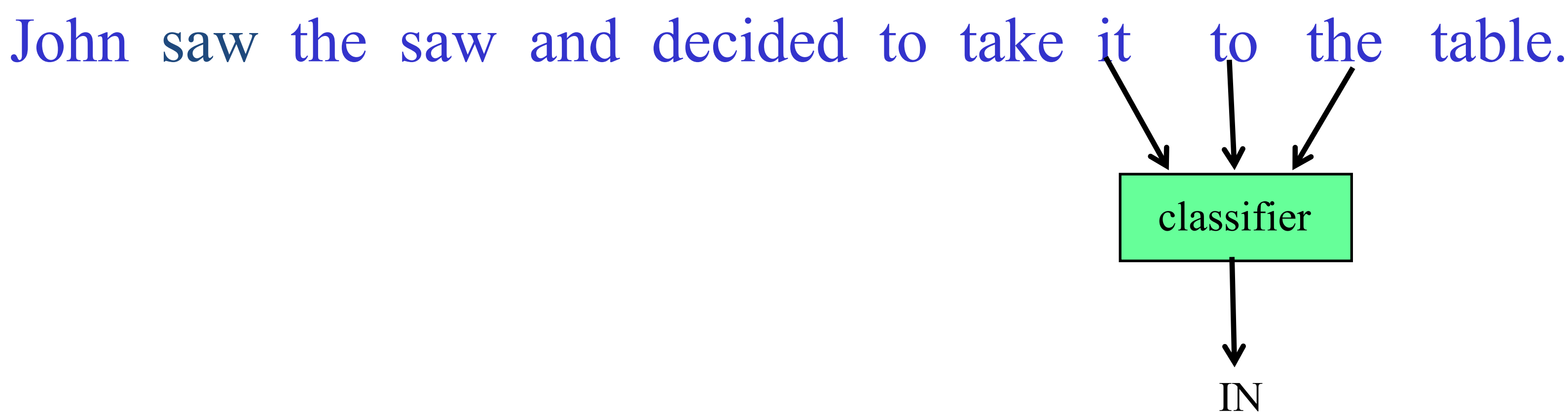
John saw the saw and decided to take it to the table.



Slide from Ray Mooney

Sequence Labeling as Classification

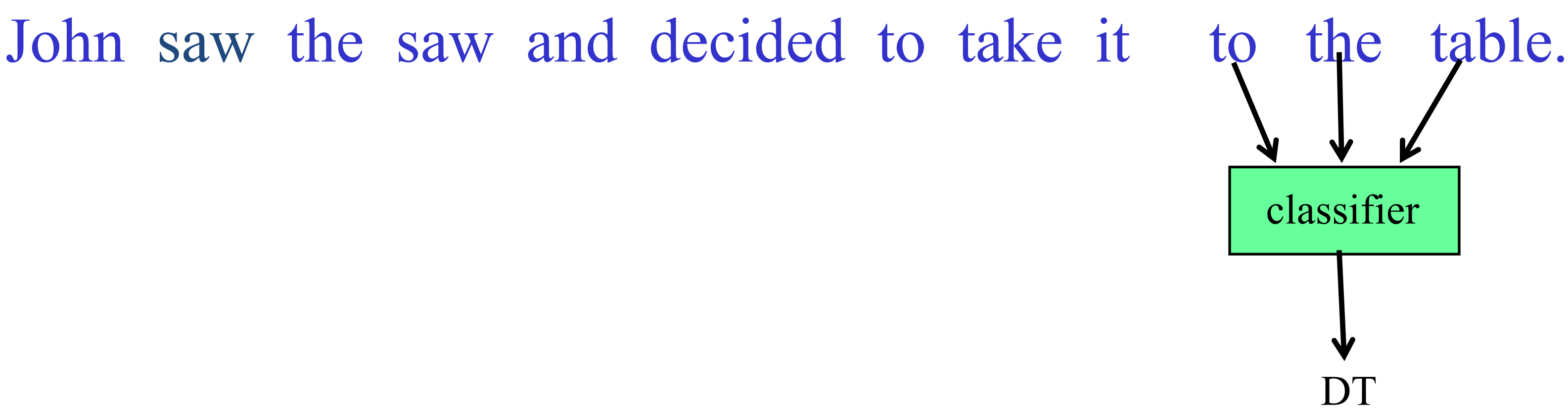
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Slide from Ray Mooney

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

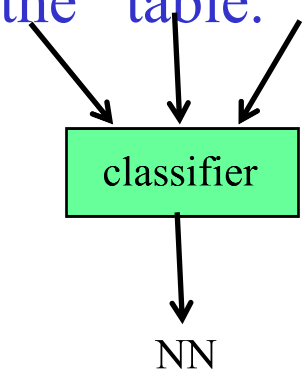


Slide from Ray Mooney

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

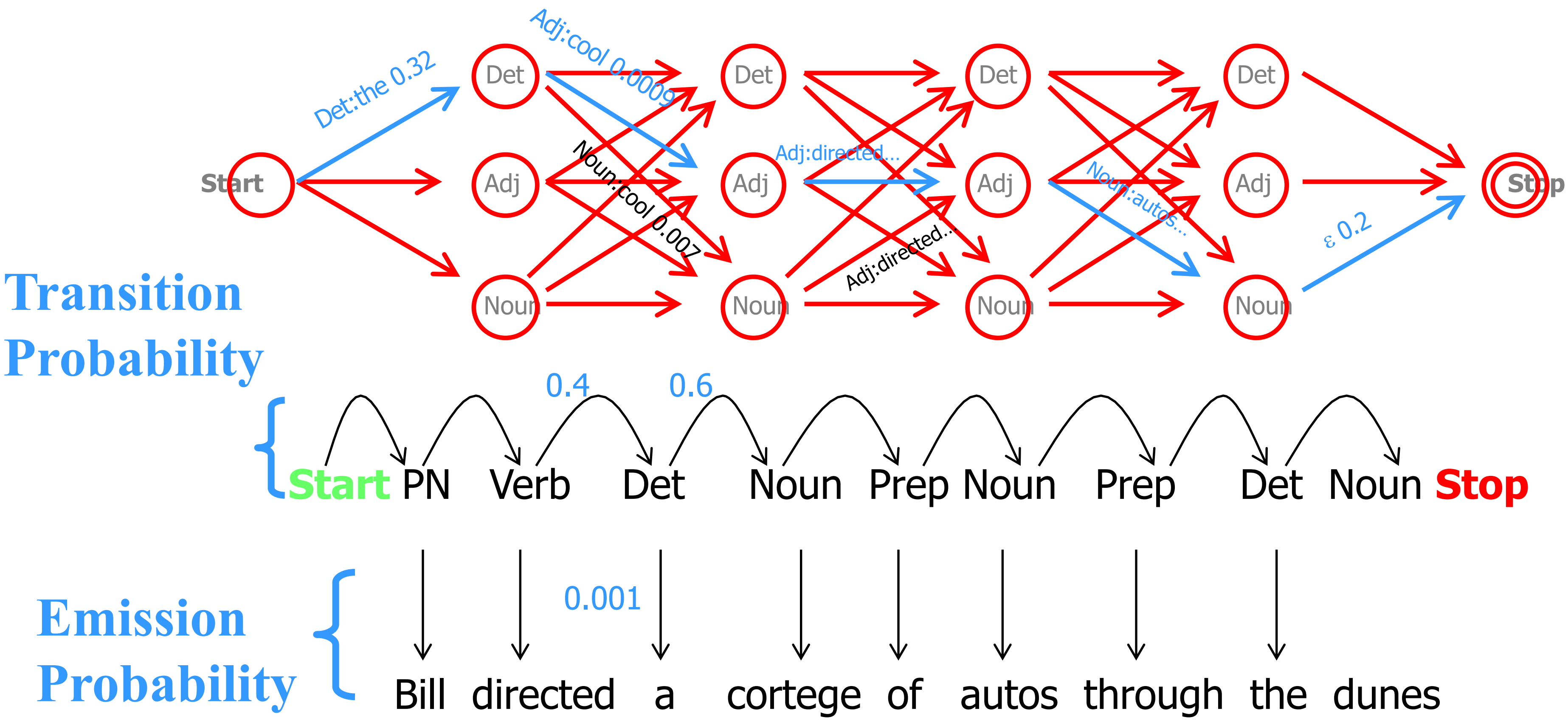
John saw the saw and decided to take it to the table.



Slide from Ray Mooney

Part of Speech Tagging

- Or we could use an HMM:



Part of Speech Tagging

- We could treat tagging as a token classification problem
  - Tag each word independently given features of context
  - And features of the word's spelling (suffixes, capitalization)

- Or we could use an HMM:
  - The point of the HMM is basically that the tag of one word might depend on the tags of adjacent words.

- Combine these two ideas??
  - We'd like rich features (e.g., in a log-linear model), but we'd also like our feature functions to depend on adjacent tags.
  - So, the problem is to predict **all** tags together.

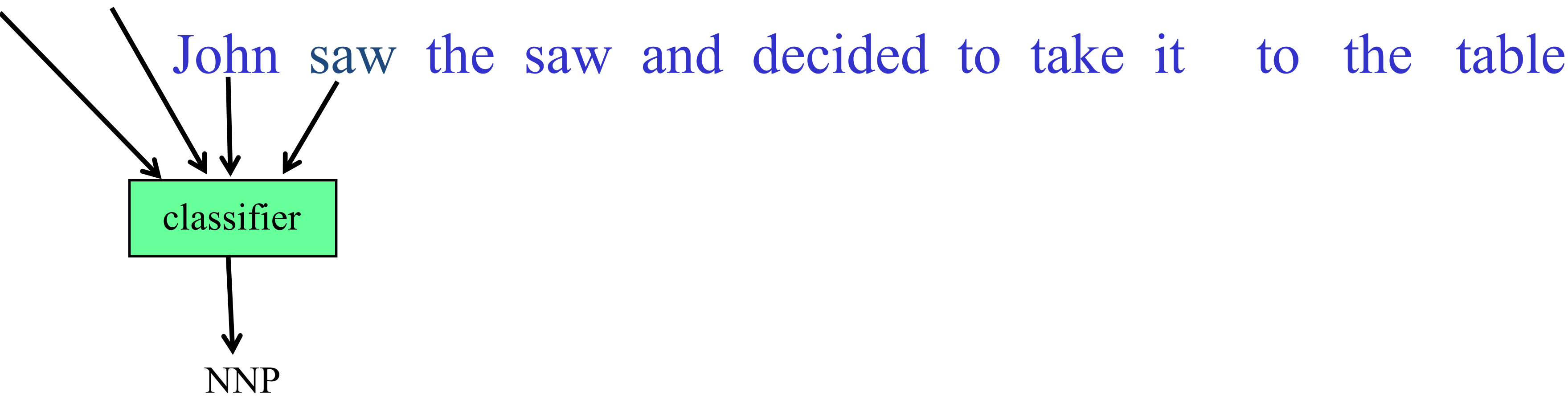
Supervised Learning Methods

- Easy to build a "yes" or "no" predictor from supervised training data
  - Plenty of software packages to do the learning & prediction
  - Lots of people in NLP never go beyond this ☺
- Harder to predict the best string or tree (set is exponentially large or infinite)

Part of Speech Tagging

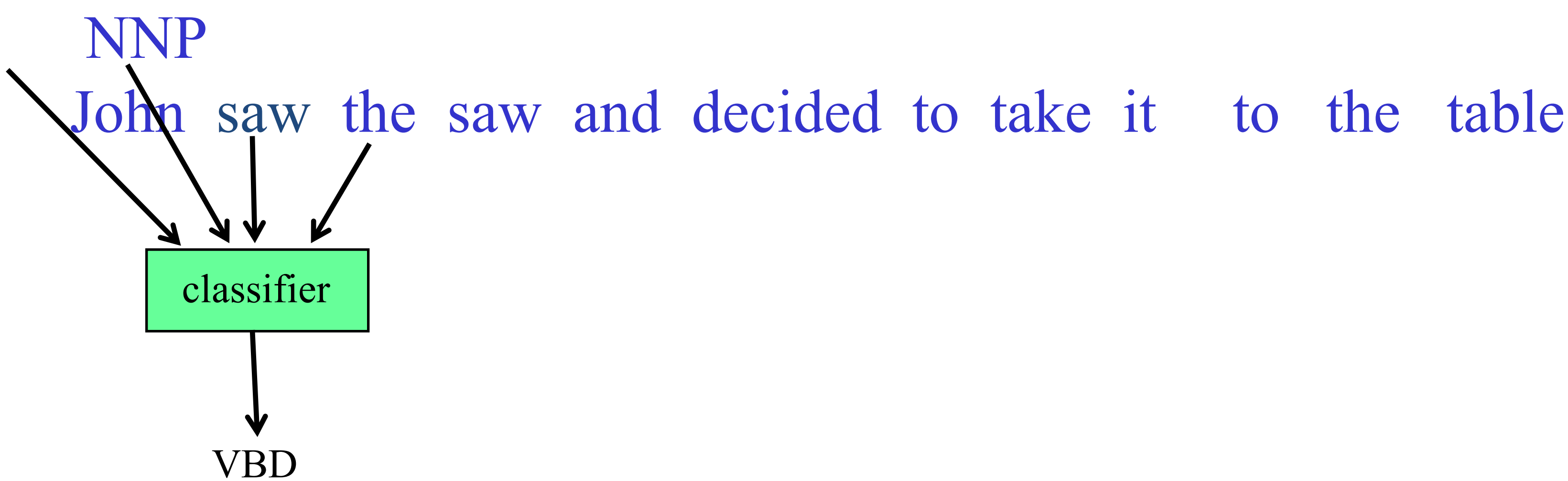
- Idea #1
  - Classify tags one at a time **from left to right**
  - Each feature function can look at the context of the word being tagged, **including the tags of all previous words**

Forward Classification



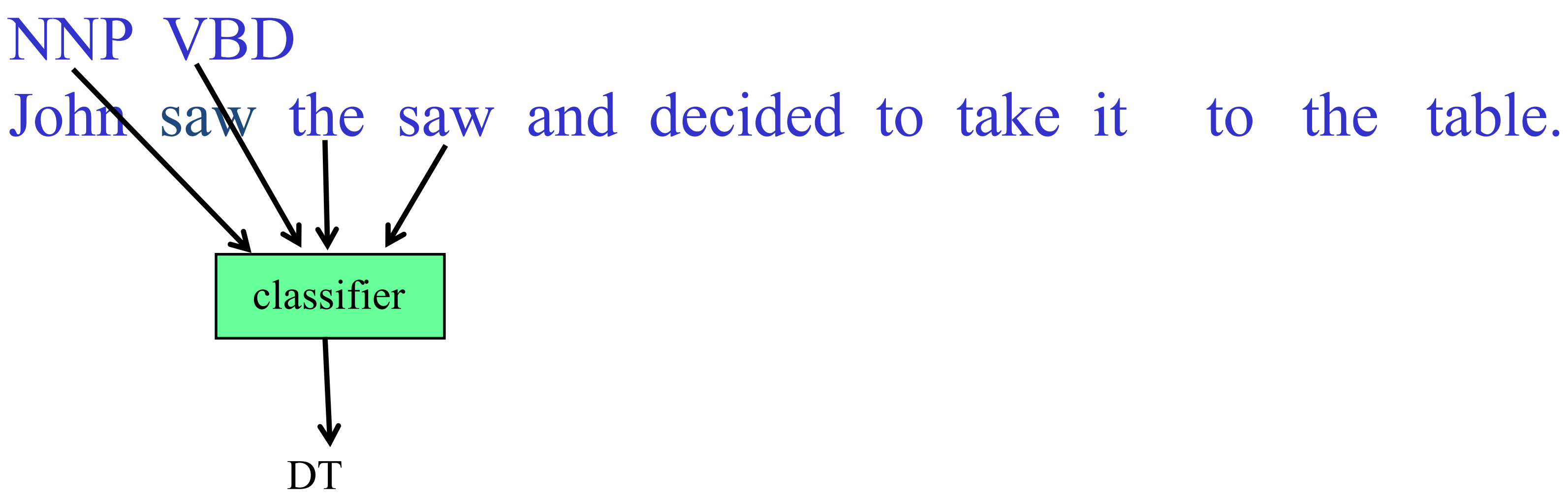
Slide from Ray Mooney

Forward Classification



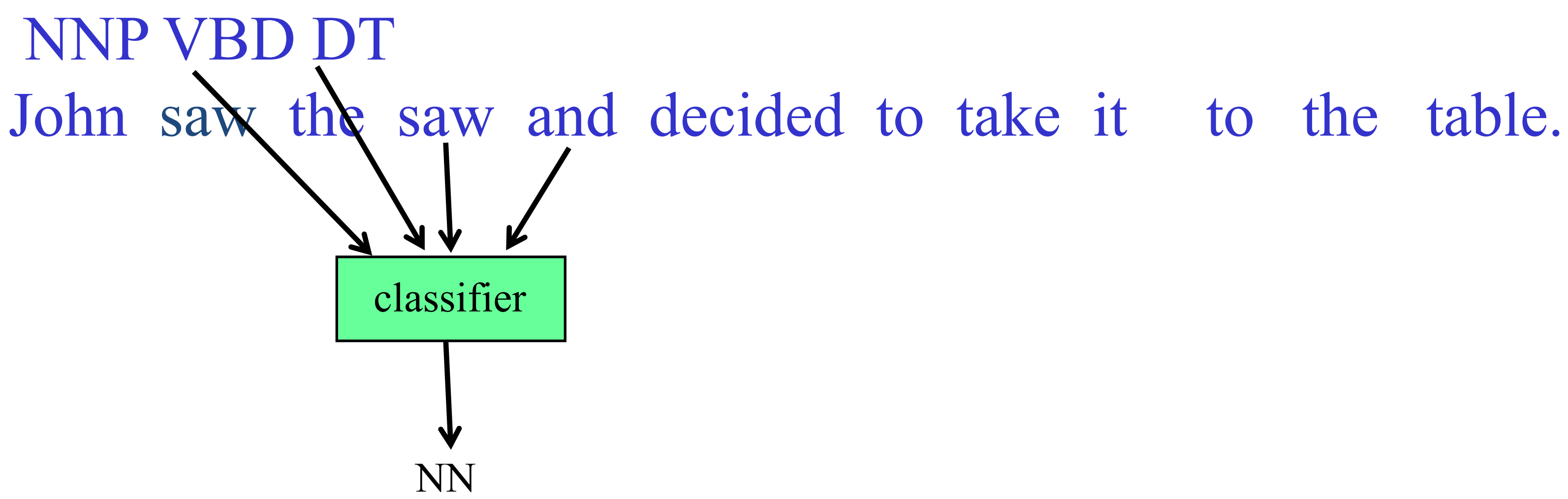
Slide from Ray Mooney

Forward Classification



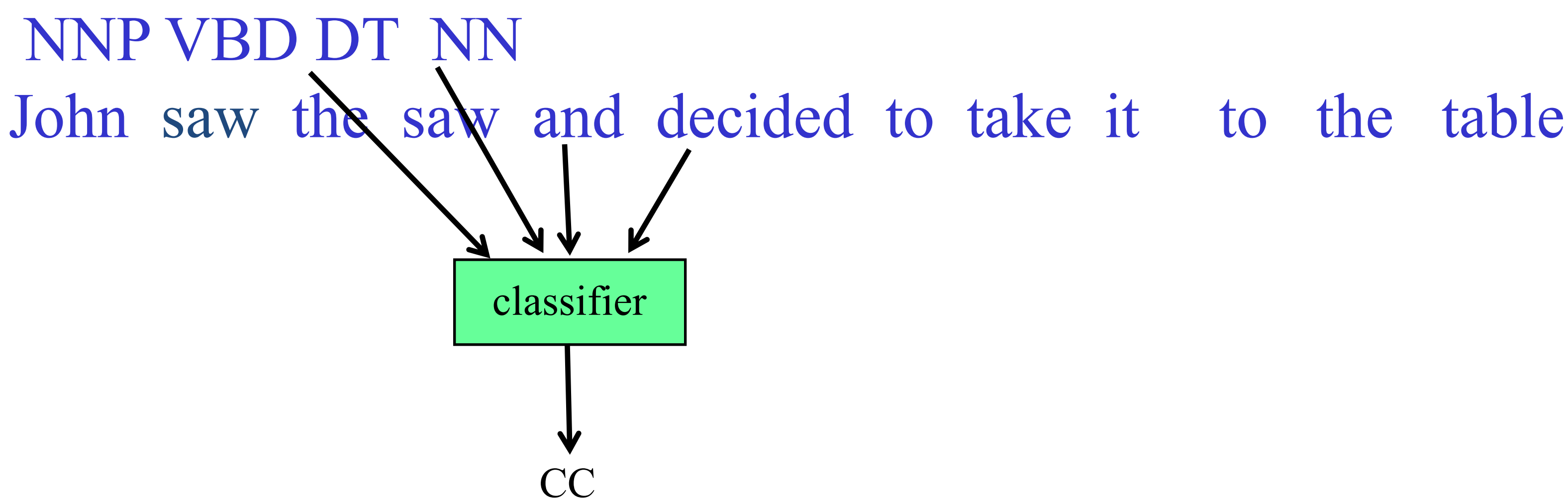
Slide from Ray Mooney

Forward Classification



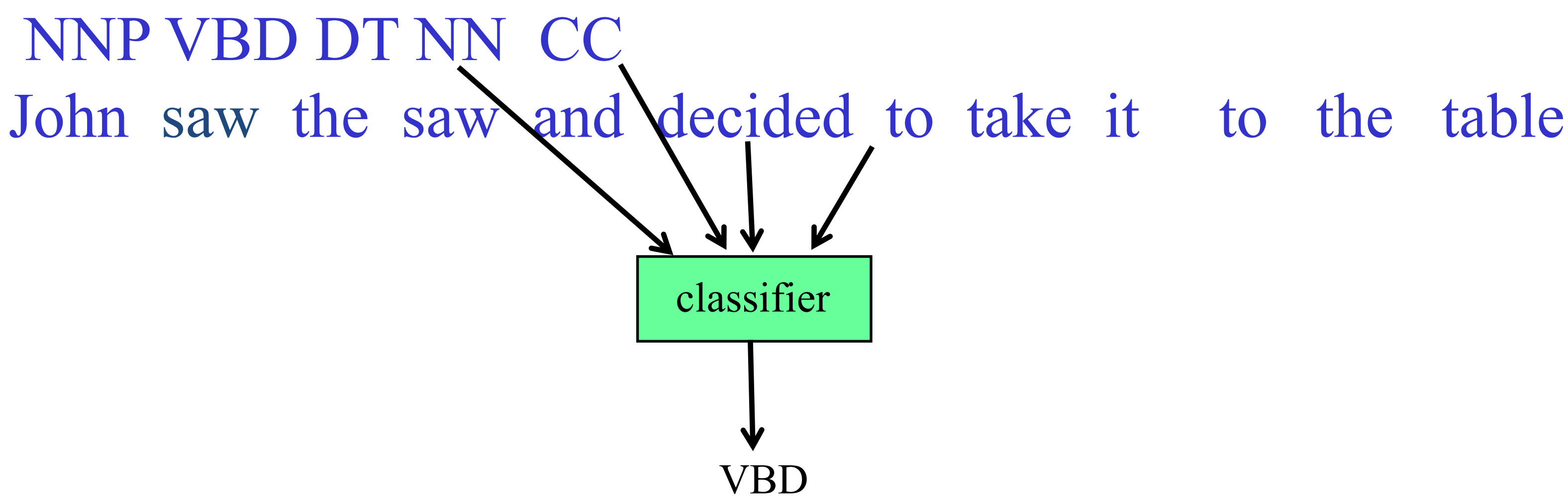
Slide from Ray Mooney

Forward Classification



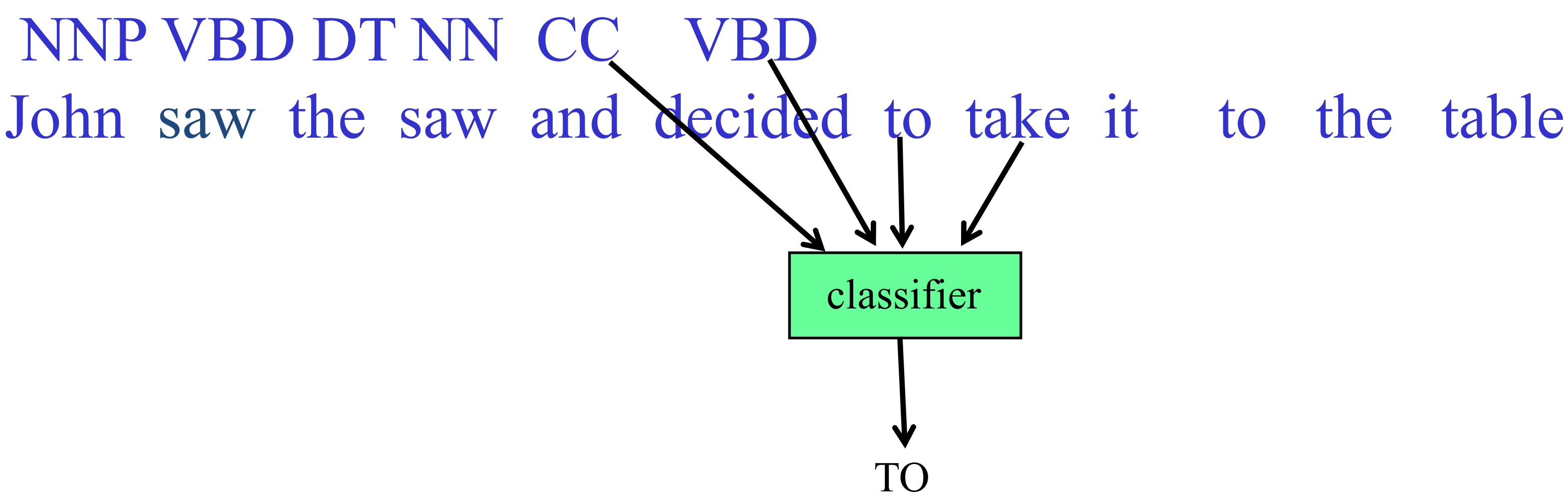
Slide from Ray Mooney

Forward Classification



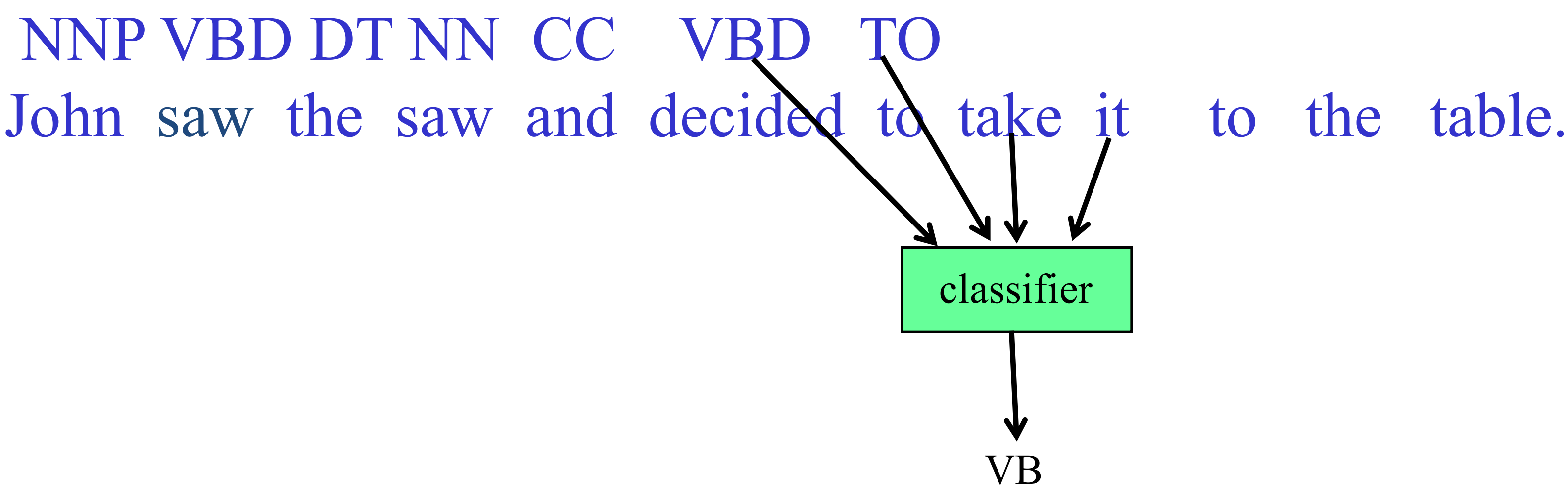
Slide from Ray Mooney

Forward Classification



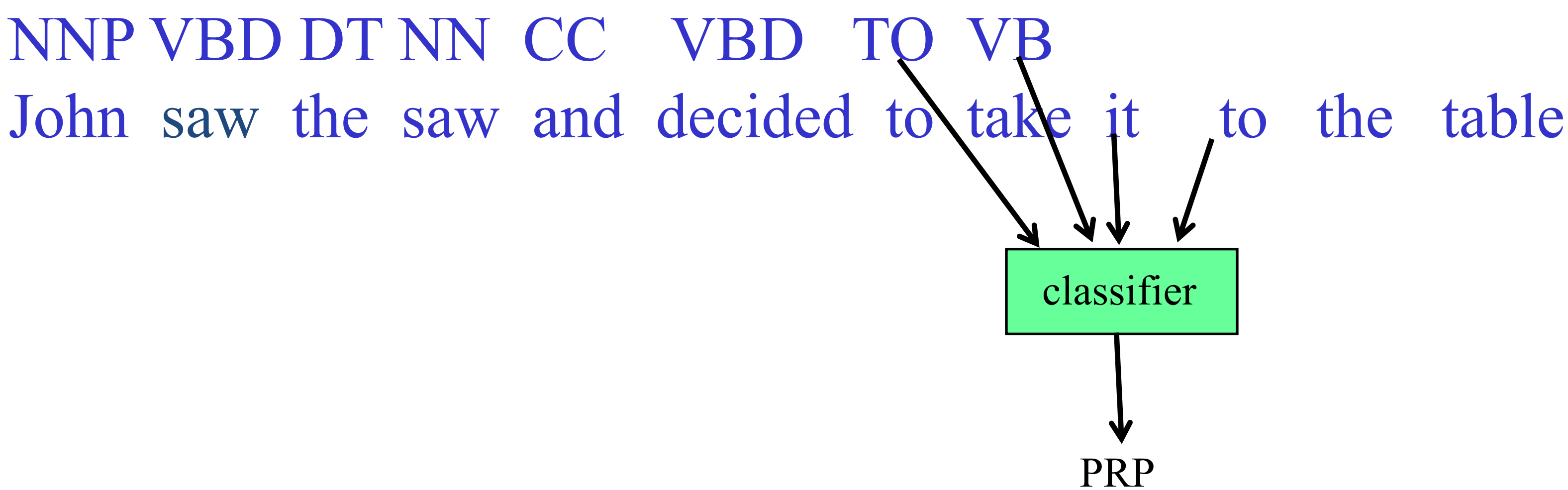
Slide from Ray Mooney

Forward Classification



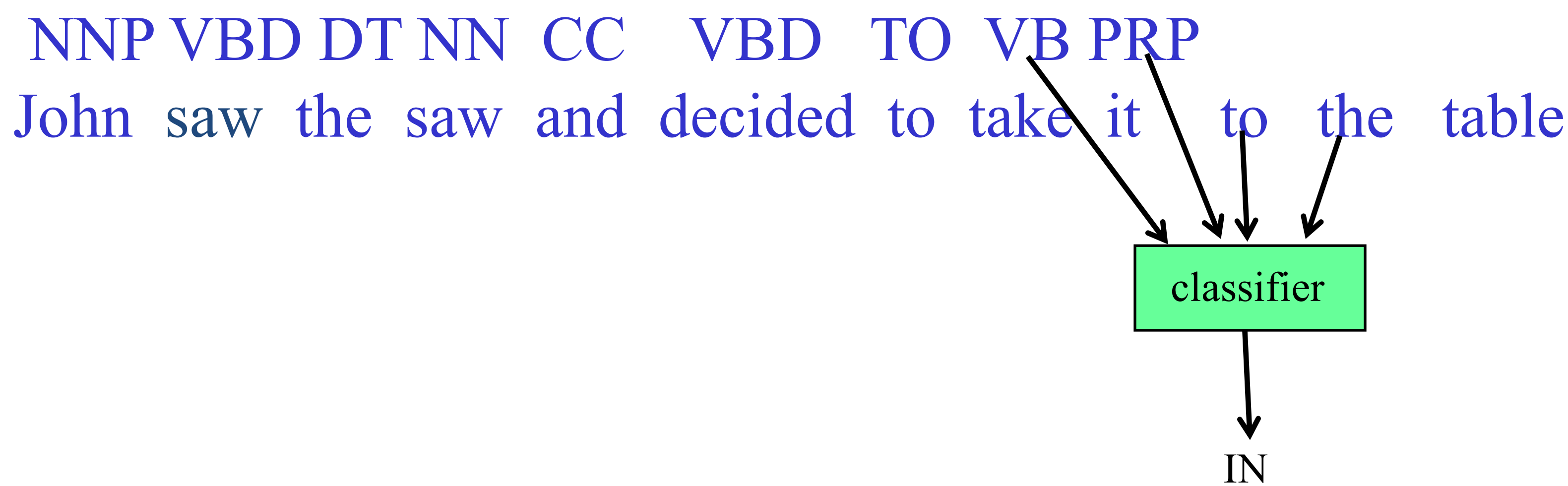
Slide from Ray Mooney

Forward Classification



Slide from Ray Mooney

Forward Classification

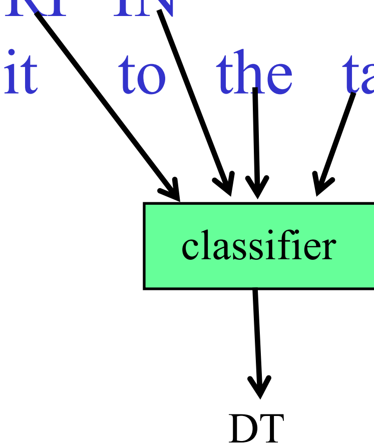


Slide from Ray Mooney

Forward Classification

NNP VBD DT NN CC VBD TO VB PRP IN

John saw the saw and decided to take it to the table.

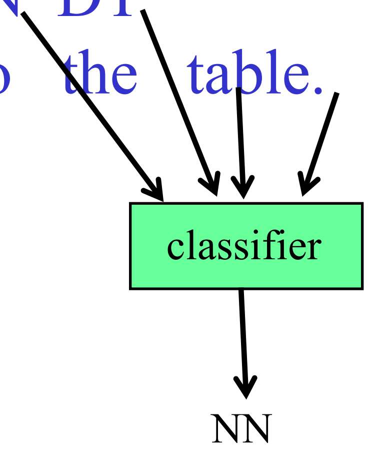


Slide from Ray Mooney

Forward Classification

NNP VBD DT NN CC VBD TO VB PRP IN DT

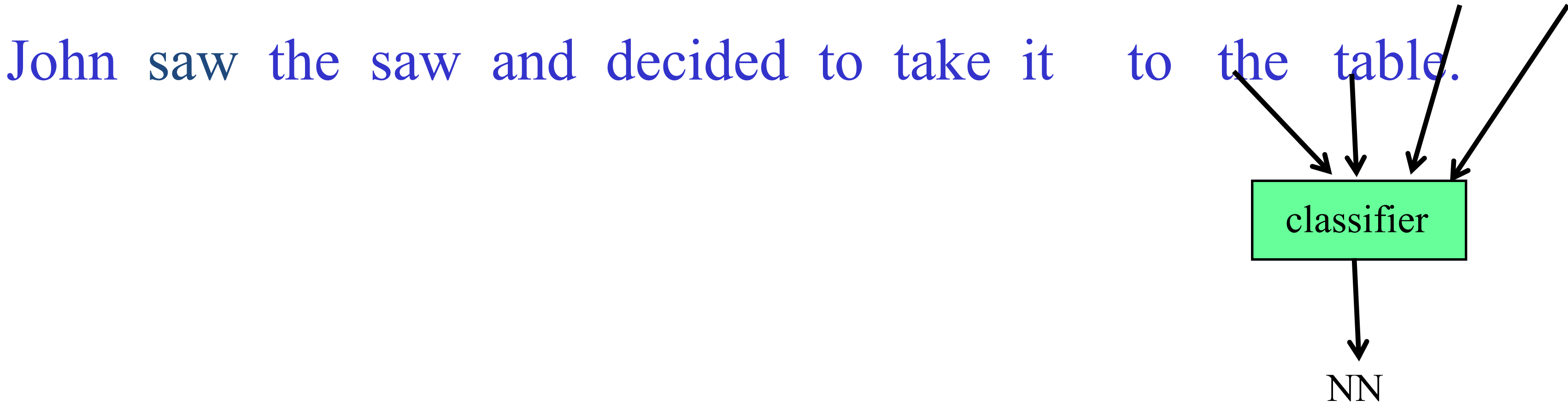
John saw the saw and decided to take it to the table.



Slide from Ray Mooney

Backward Classification

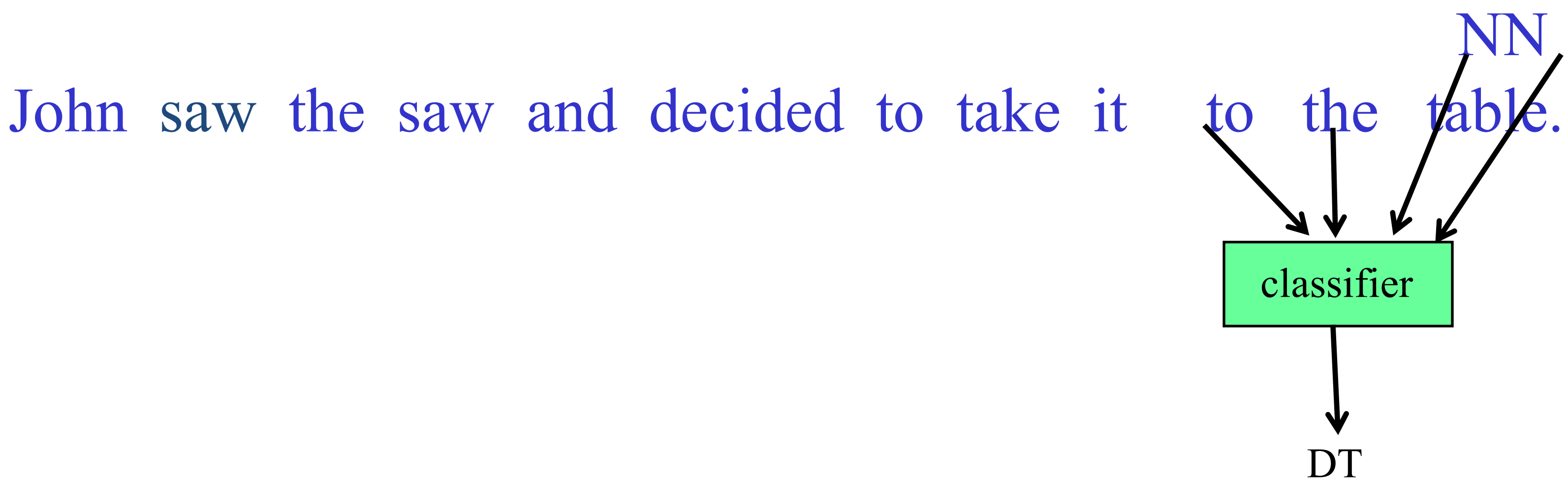
- Disambiguating "to" in this case would be even easier backward.



Slide from Ray Mooney

Backward Classification

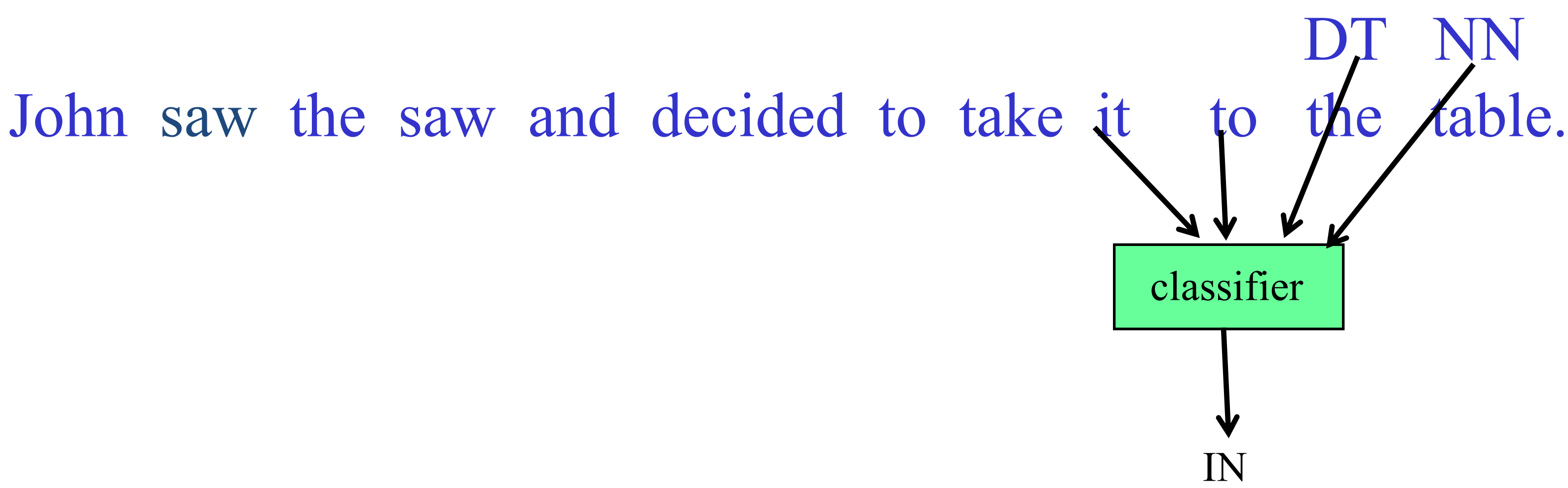
- Disambiguating "to" in this case would be even easier backward.



Slide from Ray Mooney

Backward Classification

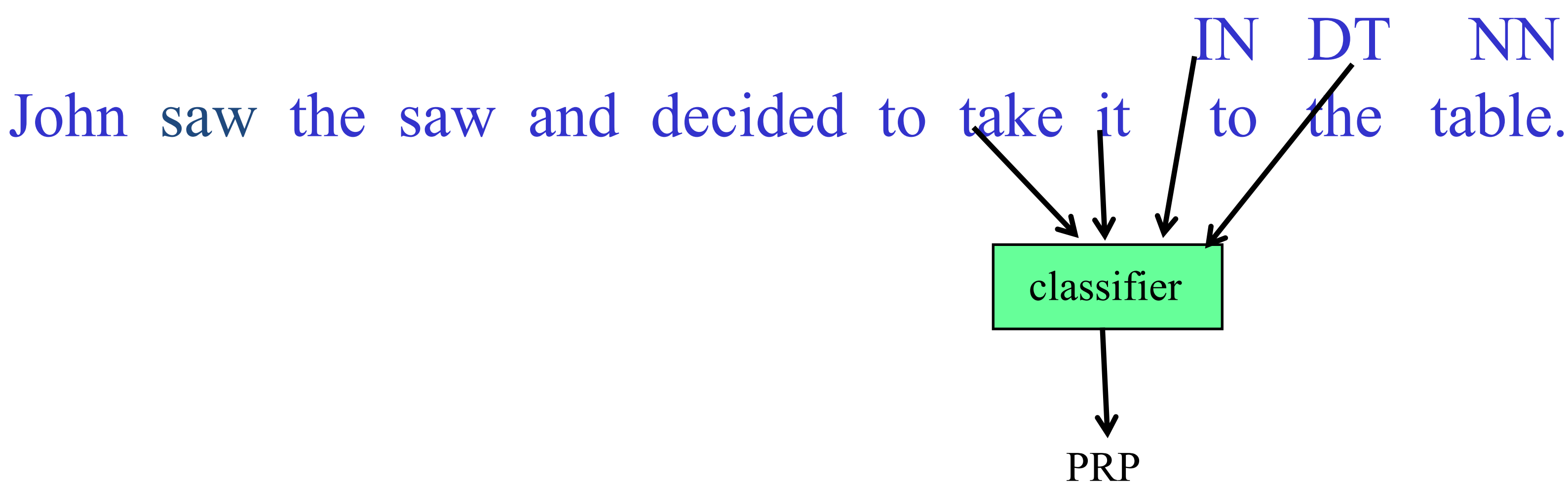
- Disambiguating "to" in this case would be even easier backward.



Slide from Ray Mooney

Backward Classification

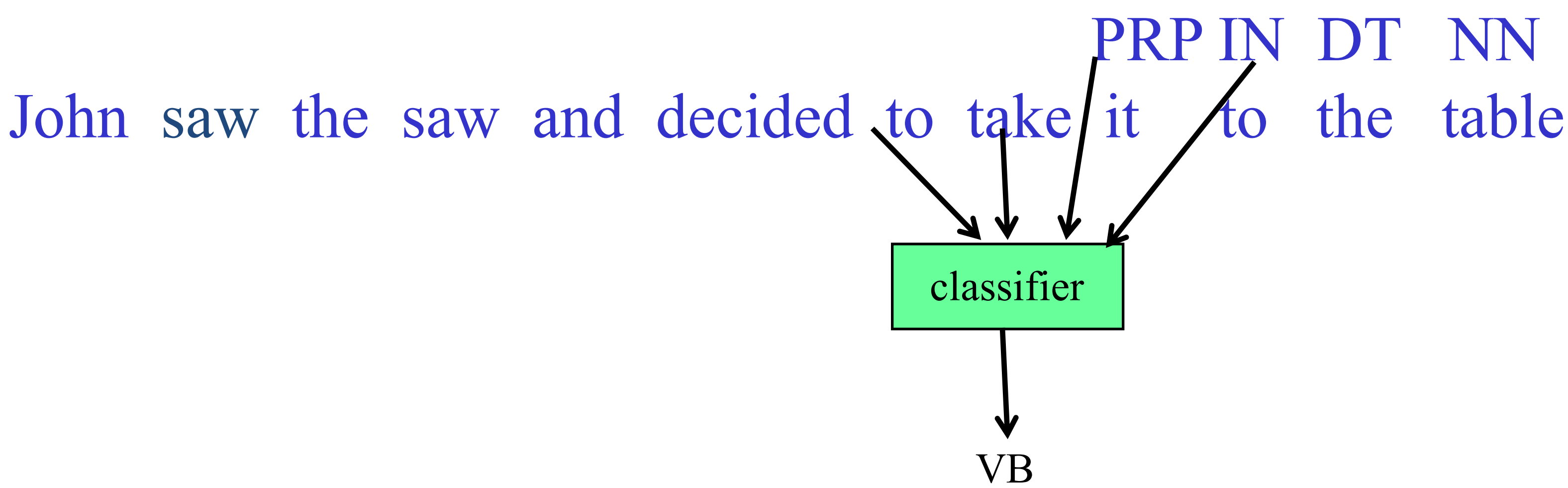
- Disambiguating "to" in this case would be even easier backward.



Slide from Ray Mooney

Backward Classification

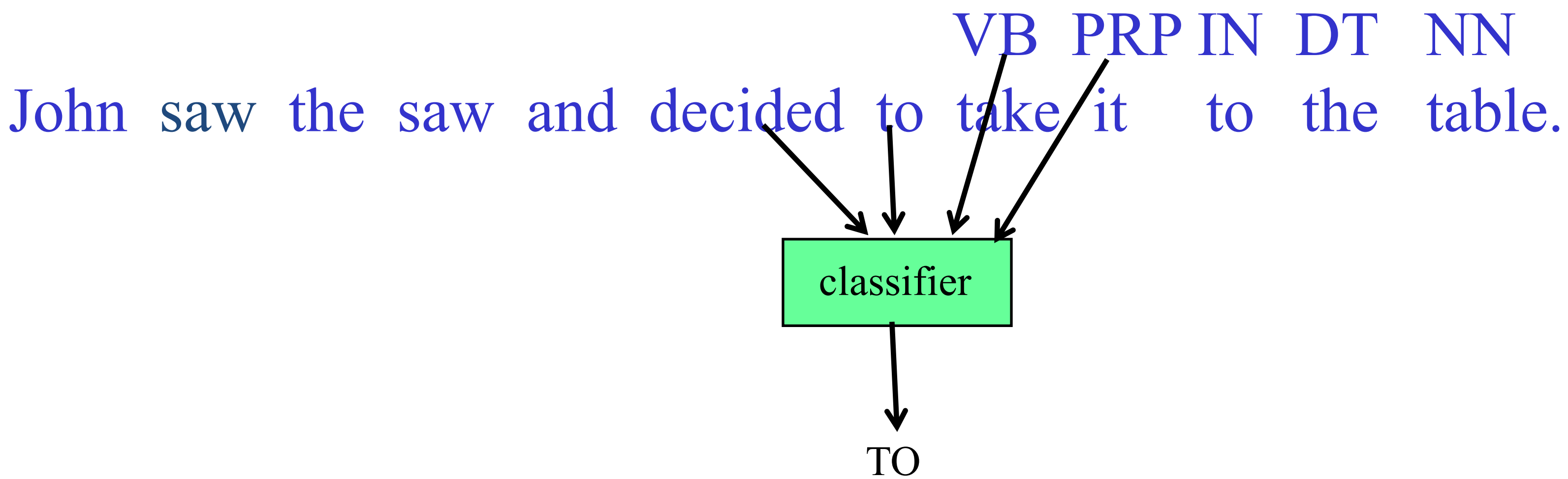
- Disambiguating "to" in this case would be even easier backward.



Slide from Ray Mooney

Backward Classification

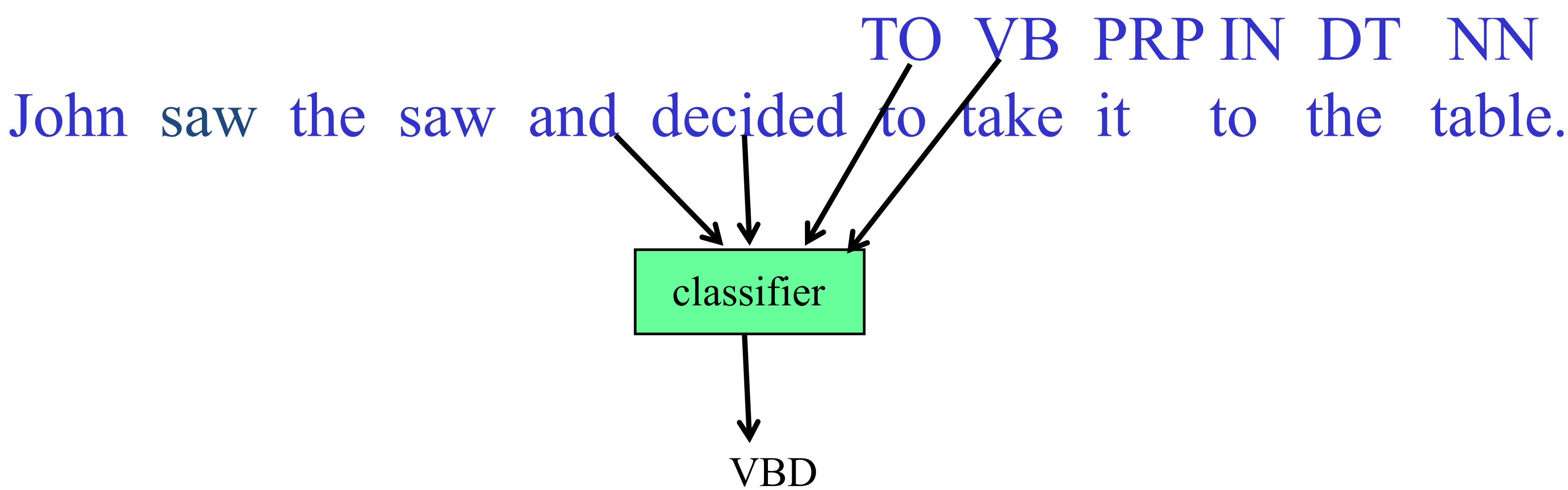
- Disambiguating "to" in this case would be even easier backward.



Slide from Ray Mooney

Backward Classification

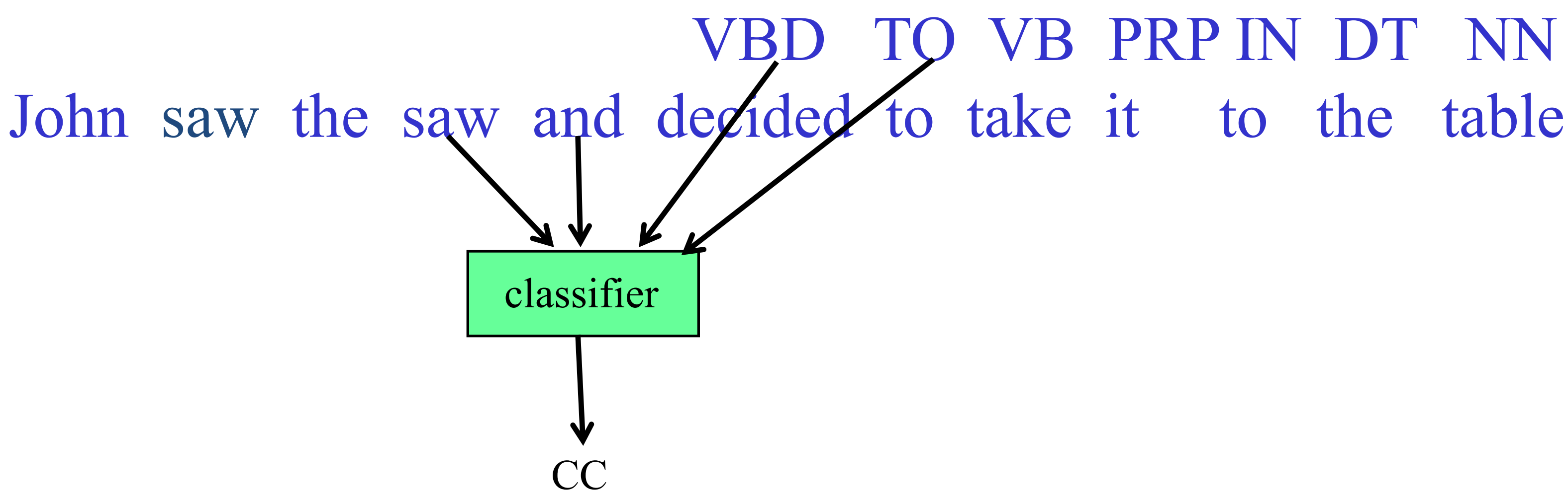
- Disambiguating "to" in this case would be even easier backward.



Slide from Ray Mooney

Backward Classification

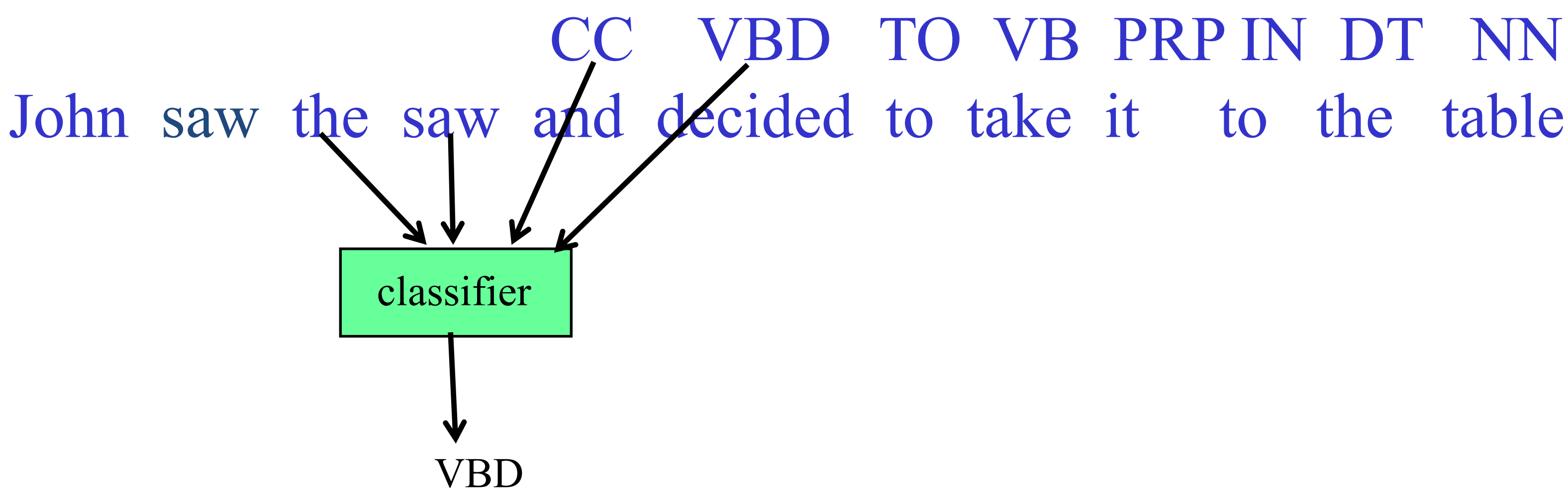
- Disambiguating "to" in this case would be even easier backward.



Slide from Ray Mooney

Backward Classification

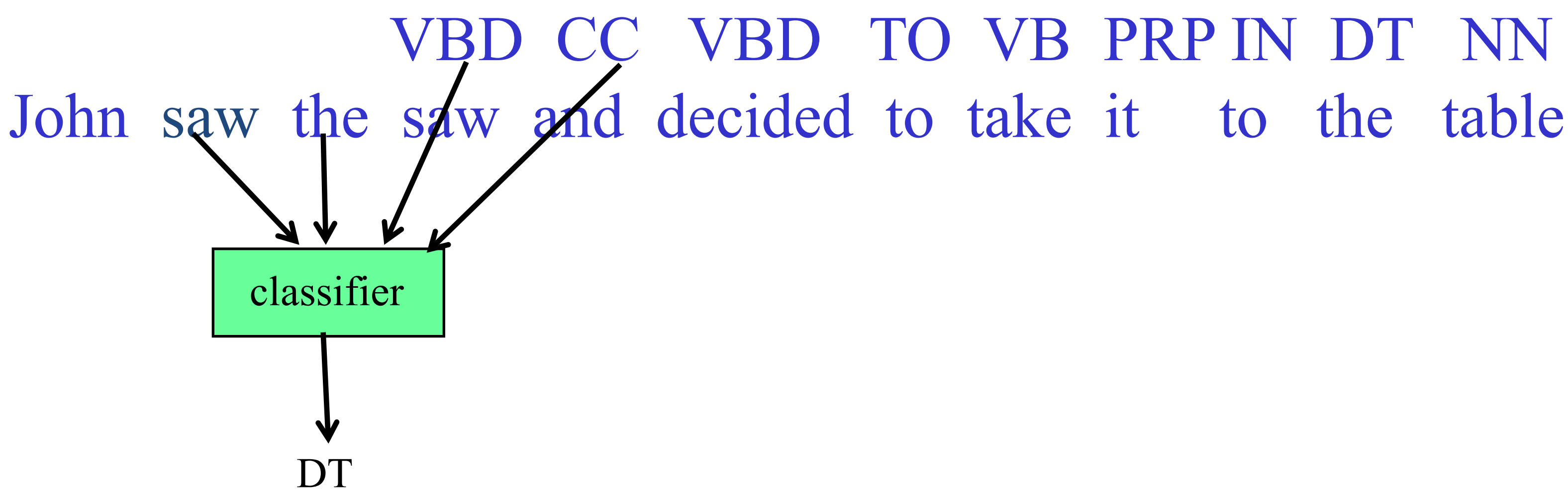
- Disambiguating "to" in this case would be even easier backward.



Slide from Ray Mooney

Backward Classification

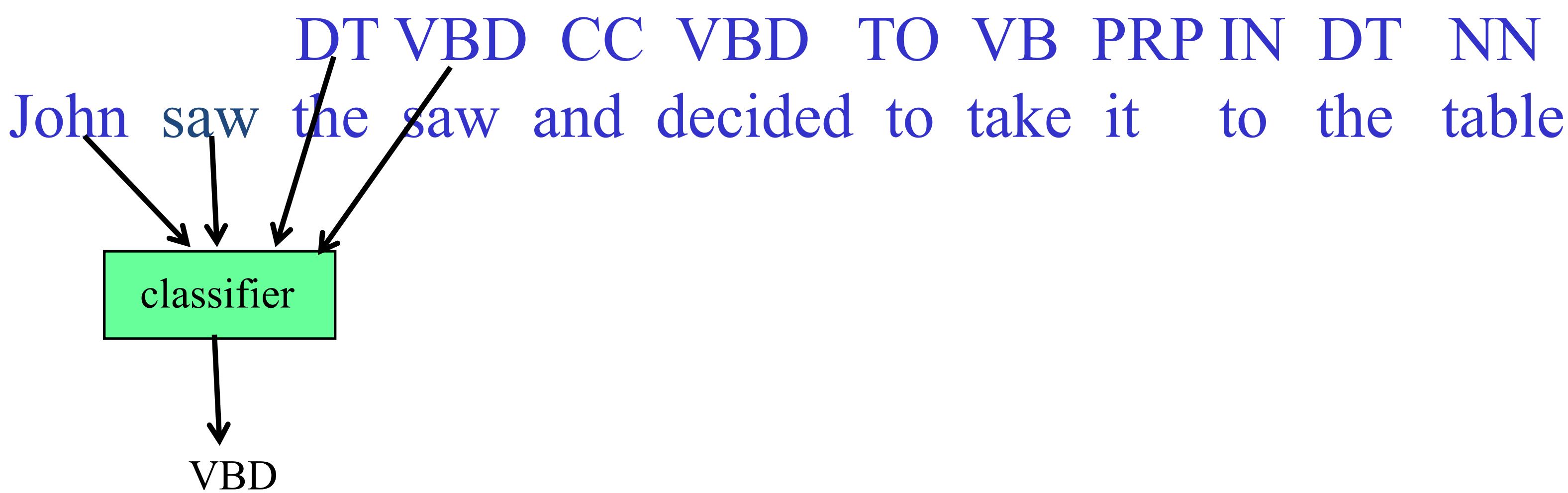
- Disambiguating "to" in this case would be even easier backward.



Slide from Ray Mooney

Backward Classification

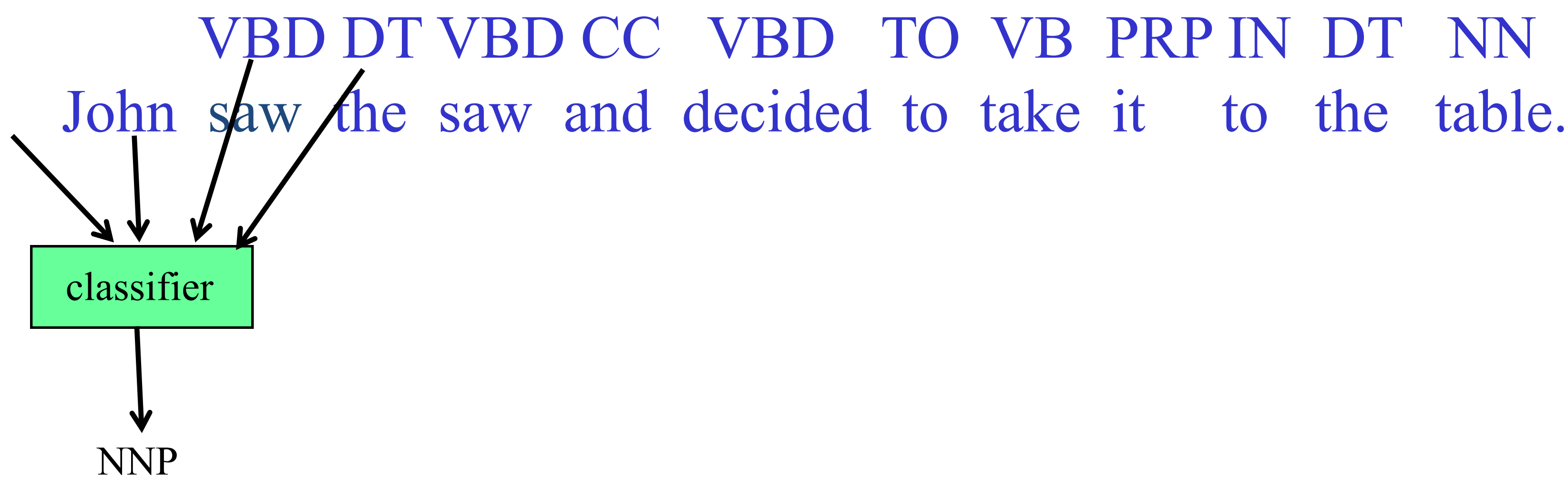
- Disambiguating "to" in this case would be even easier backward.



Slide from Ray Mooney

Backward Classification

- Disambiguating "to" in this case would be even easier backward.



Slide from Ray Mooney

Part of Speech Tagging

- Idea #1
  - Classify tags one at a time **from left to right**
    - p(tag | wordseq, prevtags) = (1/Z) exp score(tag, wordseq, prevtags)
    - where Z sums up exp score(tag', wordseq, prevtags) over all possible tags
  - Each feature function can look at the context of the word being tagged, **including the tags of all previous words**

- **Piazza Poll**:
- are there any issues with this method?

Part of Speech Tagging

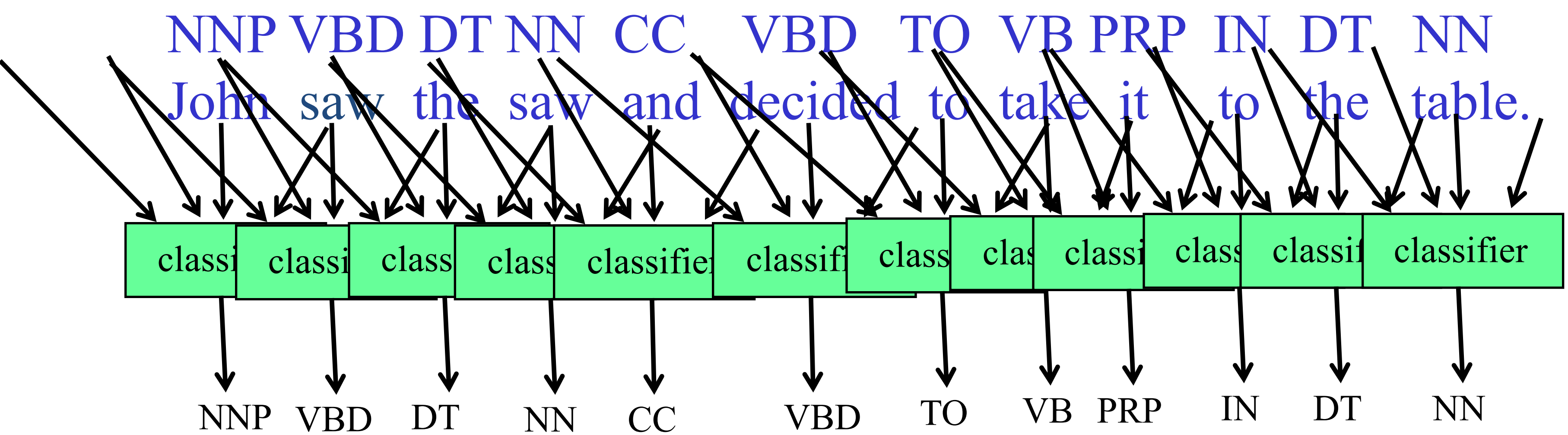
- Idea #1
  - Classify tags one at a time **from left to right**
    - p(tag | wordseq, prevtags) = (1/Z) exp score(tag, wordseq, prevtags)
    - where Z sums up exp score(tag', wordseq, prevtags) over all possible tags
  - Each feature function can look at the context of the word being tagged, **including the tags of all previous words**
  - Asymmetric: can't look at following tags, only preceding ones
- Idea #2 ("maximum entropy Markov model (MEMM)")
  - Same model, but don't **commit** to a tag before we predict the next tag. Instead, consider probabilities of all tag **sequences**.

Attack was

V

Maximum Entropy Markov Model

Is this a probable tag sequence for this sentence?



Does each of these classifiers assign a high probability to the desired tag?

Is this the most likely sequence?

(Does it maximize product of probabilities?)

Slide adapted from Ray Mooney

Part of Speech Tagging

- Idea #1
  - Classify tags one at a time **from left to right**
    - p(tag | wordseq, prevtags) = (1/Z) exp score(tag, wordseq, prevtags)
    - where Z sums up exp score(tag', wordseq, prevtags) over all possible tags
  - Each feature function can look at the context of the word being tagged, **including the tags of all previous words**
  - Asymmetric: can't look at following tags, only preceding ones
- Idea #2 ("maximum entropy Markov model (MEMM)")
  - Same model, but don't **commit** to a tag before we predict the next tag. Instead, consider probabilities of all tag **sequences**.
  - Use dynamic programming to find the most probable sequence
    - For dynamic programming to work, features can only consider the (n-1) previous tags, just as in an HMM
    - Same algorithms as in an HMM, but now transition probability is p(tag | previous n-1 tags **<u>and all words</u>**)
  - Still asymmetric: can't look at following tags

Part of Speech Tagging

- Idea #1
  - Classify tags one at a time **from left to right**
    - p(tag | wordseq, prevtags) = (1/Z) exp score(tag, wordseq, prevtags)
    - where Z sums up exp score(tag', wordseq, prevtags) over all possible tags
- Idea #2 ("maximum entropy Markov model (MEMM)")
  - Same model, but don't **commit** to a tag before we predict the next tag. Instead, evaluate probability of every tag sequence.
- Idea #3 ("linear-chain conditional random field (CRF)")
  - This version is symmetric, and very popular.
  - Score each tag sequence as a whole, using arbitrary features
    - p(tagseq | wordseq) = (1/Z) exp score(tagseq, wordseq)
    - where Z sums up exp score(tagseq', wordseq) over competing tagseqs
  - Can still compute Z and best path using dynamic programming

Supervised Learning Methods

- Easy to build a "yes" or "no" predictor from supervised training data
  - Plenty of software packages to do the learning & prediction
  - Lots of people in NLP never go beyond this ☺
- Similarly, easy to build a system that chooses from a small finite set
  - Basically the same deal
- Harder to predict the best string or tree (set is exponentially large or infinite)
  - Requires dynamic programming; you might have to write your own code
  - But finite-state or CRF toolkits will find the best string for you
  - For recurrent neural networks, you still need to consider decoding algorithms…

5-minute break

NLP Tasks

1. Classify the entire document
2. Classify individual word tokens
3. Identify phrases (“chunking”)

Named Entity Recognition

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 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.

NE Types

Type	Tag	Sample Categories
People	PER	Individuals, fictional characters, small groups
Organization	ORG	Companies, agencies, political parties, religious groups, sports teams
Location	LOC	Physical extents, mountains, lakes, seas
Geo-Political Entity	GPE	Countries, states, provinces, counties
Facility	FAC	Bridges, buildings, airports
Vehicles	VEH	Planes, trains, and automobiles

Type	Example
People	*Turing* is often considered to be the father of modern computer science.
Organization	The *IPCC* said it is likely that future tropical cyclones will become more intense.
Location	The *Mt. Sanitas* loop hike begins at the base of *Sunshine Canyon*.
Geo-Political Entity	*Palo Alto* is looking at raising the fees for parking in the University Avenue district.
Facility	Drivers were advised to consider either the *Tappan Zee Bridge* or the *Lincoln Tunnel*.
Vehicles	The updated *Mini Cooper* retains its charm and agility.

Information Extraction

As a task: **Filling slots in a database from sub-segments of text.**

October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access.“

Richard Stallman, founder of the Free Software Foundation, countered saying…

IE

NAME	TITLE	ORGANIZATION
Bill Gates	CEO	Microsoft
Bill Veghte	VP	Microsoft
Richard Stallman	founder	Free Soft..

Slide from Chris Brew, adapted from slide by William Cohen

The Semantic Web

- A simple scheme for representing factual knowledge as a labeled graph
  - [draw example with courses, students, their names and locations, etc.]
- Many information extraction tasks aim to produce something like this
- Is a labeled graph (triples) really enough?
  - ☺ Can transform k-tuples to triples
    (cf. Davidsonian event variable)
  - ☹ Supports facts about individuals, but no direct support for quantifiers or reasoning

Phrase Types to Identify for IE

Closed set

U.S. states

He was born in Alabama…

The big Wyoming sky…

Regular set

U.S. phone numbers

Phone: (413) 545-1323

The CALD main office can be reached at 412-268-1299

Complex pattern

U.S. postal addresses

University of Arkansas
P.O. Box 140
Hope, AR 71802

Headquarters:
1128 Main Street, 4th Floor
Cincinnati, Ohio 45210

Ambiguous patterns, needing context and many sources of evidence

Person names

…was among the six houses sold by Hope Feldman that year.

Pawel Opalinski, Software Engineer at WhizBang Labs.

Slide from Chris Brew, adapted from slide by William Cohen

Identifying phrases

- A key step in IE is to identify relevant phrases
  - Named entities
    - As on previous slides
  - Relationship phrases
    - "said", "according to", …
    - "was born in", "hails from", …
    - "bought", "hopes to acquire", "formed a joint agreement with", …
  - Simple syntactic chunks (e.g., non-recursive NPs)
    - "Syntactic chunking" sometimes done before (or instead of) parsing
    - Also, "segmentation": divide Chinese text into words (no spaces)
- So, how do we learn to mark phrases?
  - Earlier, we built an FST to mark dates by inserting brackets
  - But, it's common to set this up as a tagging problem …

Reduce to a tagging problem …

- The IOB encoding (Ramshaw & Marcus 1995):
  - B_X = "beginning" (first word of an X)
  - I_X = "inside" (non-first word of an X)
  - O = "outside" (not in any phrase)
  - Does not allow overlapping or recursive phrases

…	United	Nations	said	Friday	it	has	increased	…
	B_ORG	I_ORG	O	O	O	O	O	

…	the	move	,	spokesman	Tim	Wagner	said	…
	O	O	O	O	B_PER	I_PER	O	

Slide adapted from Chris Brew

Some Simple NER Features

POS tags and chunks from earlier processing

Now predict NER tagseq

Features				Label
American	NNP	B_{NP}	cap	B_{ORG}
Airlines	NNPS	I_{NP}	cap	I_{ORG}
,	PUNC	O	punc	O
a	DT	B_{NP}	lower	O
unit	NN	I_{NP}	lower	O
of	IN	B_{PP}	lower	O
AMR	NNP	B_{NP}	upper	B_{ORG}
Corp.	NNP	I_{NP}	cap_punc	I_{ORG}
,	PUNC	O	punc	O
immediately	RB	B_{ADVP}	lower	O
matched	VBD	B_{VP}	lower	O
the	DT	B_{NP}	lower	O
move	NN	I_{NP}	lower	O
,	PUNC	O	punc	O
spokesman	NN	B_{NP}	lower	O
Tim	NNP	I_{NP}	cap	B_{PER}
		I_{NP}	cap	I_{PER}
		B_{VP}	lower	O
		O	punc	O

A feature of this tagseq might give a positive or negative weight to this B_ORG in conjunction with some subset of the nearby properties

Or even faraway properties: B_ORG is more likely in a sentence with a spokesman!

Slide adapted from Jim Martin

Example applications for IE

- Classified ads
- Restaurant reviews
- Bibliographic citations
- Appointment emails
- Legal opinions
- Papers describing clinical medical studies
- …

NLP Tasks

1. Classify the entire document
2. Classify individual word tokens
3. Identify phrases (“chunking”)
4. Syntactic annotation (parsing)

Parser Evaluation Metrics

- Runtime
- Exact match
  - Is the parse 100% correct?
- Labeled precision, recall, F-measure of constituents
  - Precision: You predicted (NP,5,8); was it right?
  - Recall: (NP,5,8) was right; did you predict it?
- Easier versions:
  - Unlabeled: Don't worry about getting (NP,5,8) right, only (5,8)
  - Short sentences: Only test on sentences of ≤ 15, ≤ 40, ≤ 100 words
  - Dependency parsing: Labeled and unlabeled attachment accuracy
- Crossing brackets
  - You predicted (…,5,8), but there was really a constituent (…,6,10)

Labeled Dependency Parsing

Raw sentence

He reckons the current account deficit will narrow to only 1.8 billion in September.

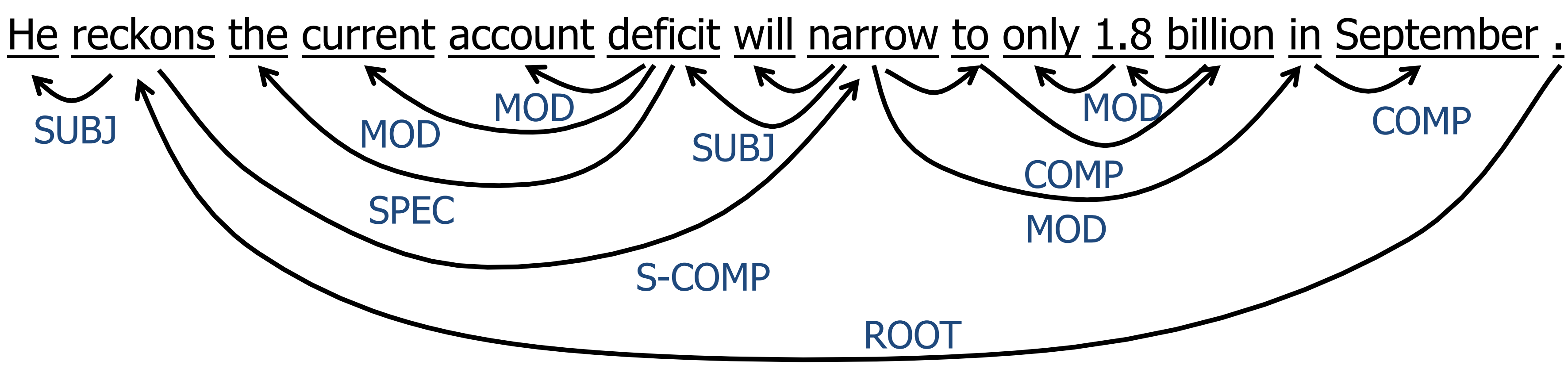
Part-of-speech tagging

POS-tagged sentence

He	reckons	the	current	account	deficit	will	narrow	to	only	1.8	billion	in	September	.
PRP	VBZ	DT	JJ	NN	NN	MD	VB	TO	RB	CD	CD	IN	NNP	.

Word dependency parsing

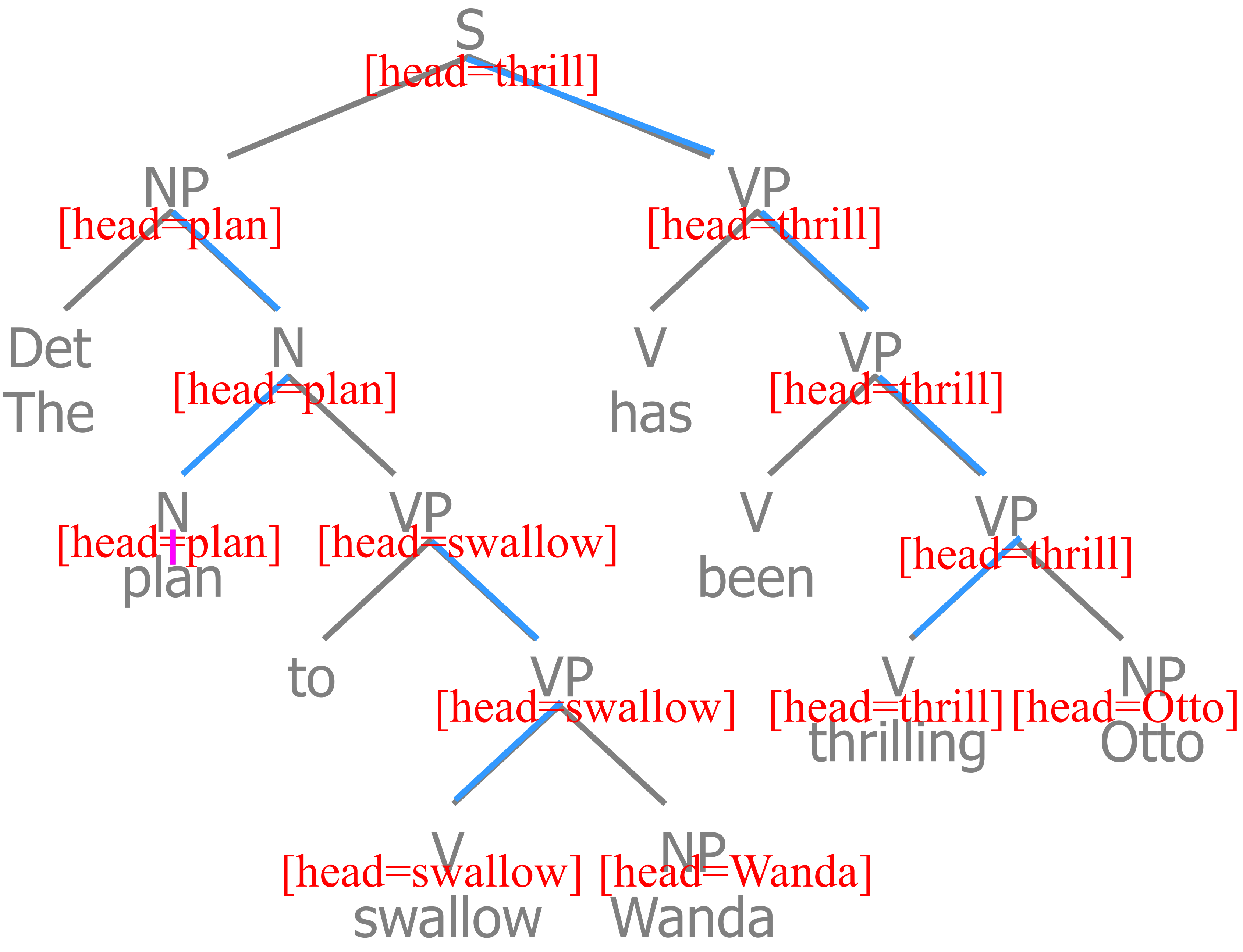
Word dependency parsed sentence



slide adapted from Yuji Matsumoto

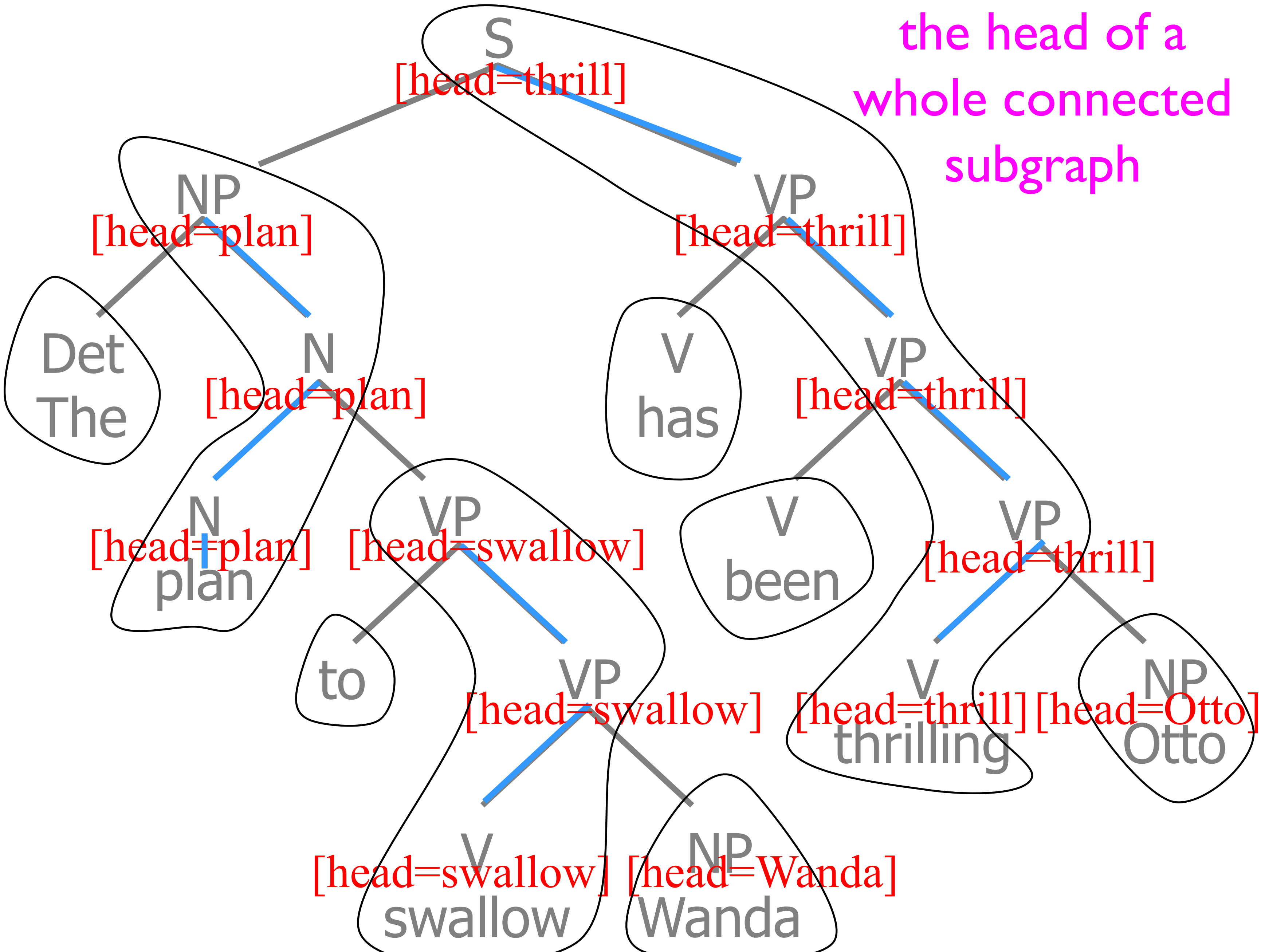
Dependency Trees

1. Assign heads



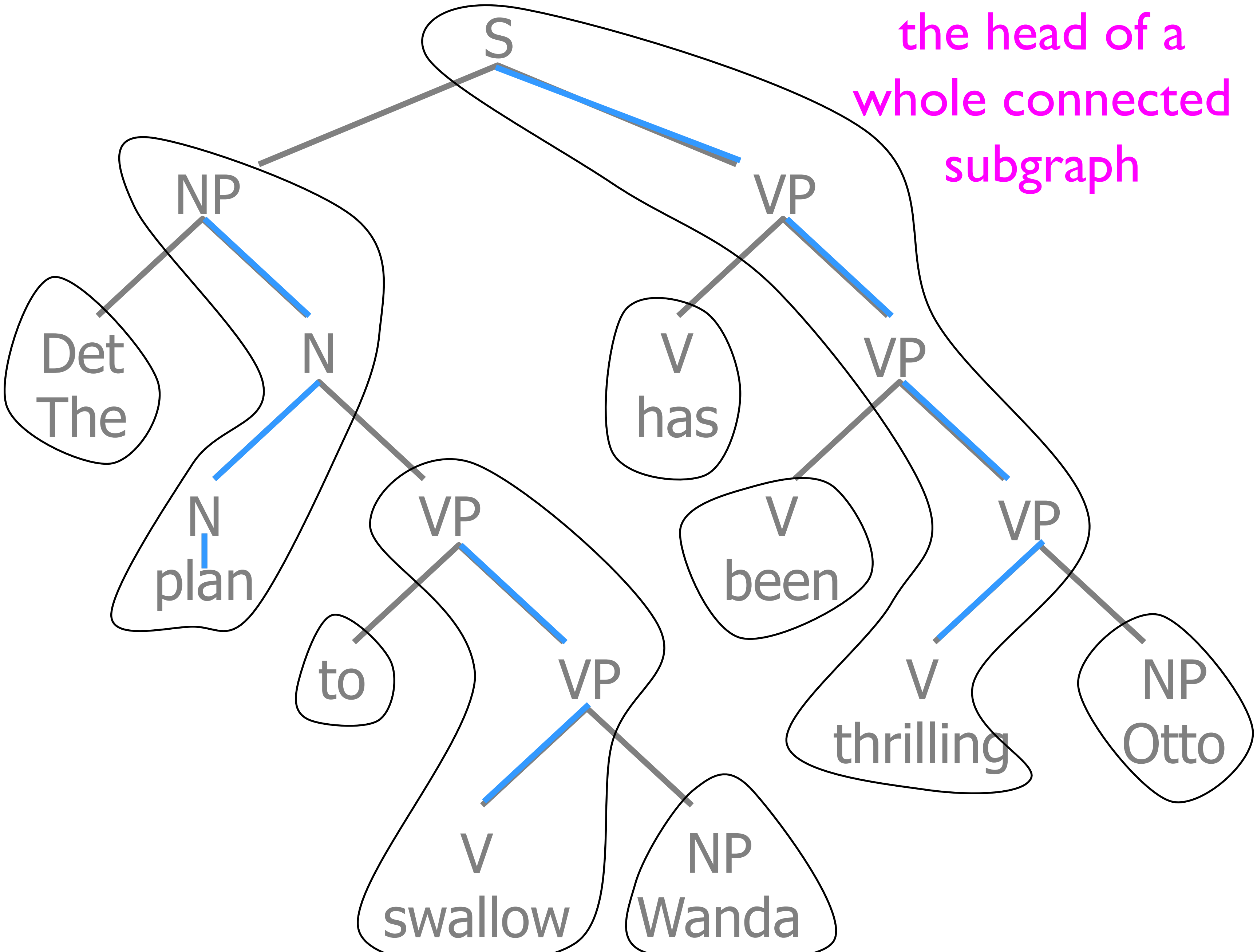
Dependency Trees

2. Each word is the head of a whole connected subgraph



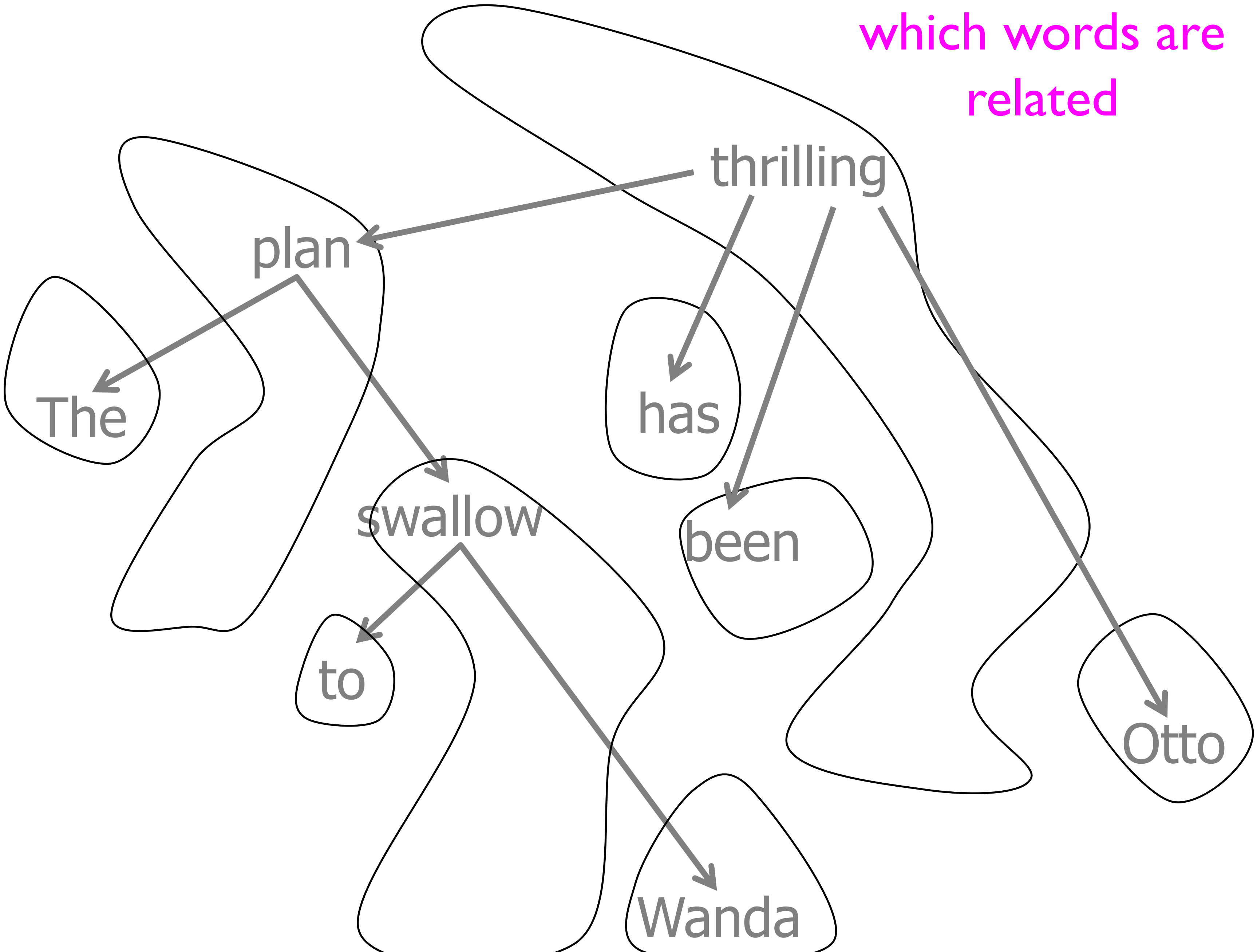
Dependency Trees

2. Each word is the head of a whole connected subgraph



Dependency Trees

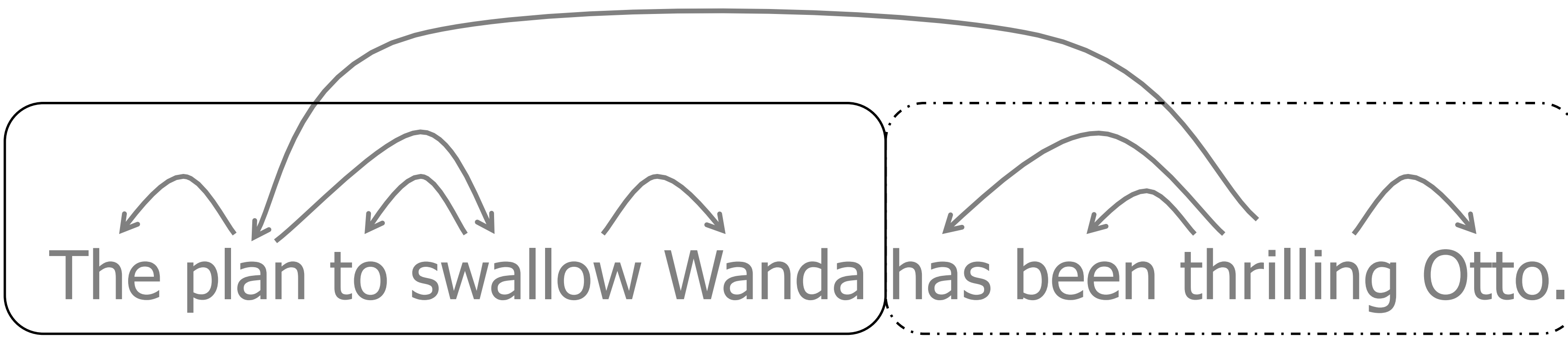
3. Just look at which words are related



Dependency Trees

4. Optionally flatten the drawing

- Shows which words modify ("depend on") another word
- Each subtree of the dependency tree is still a constituent
  - But not all of the original constituents are subtrees (e.g., VP)



- Easy to spot semantic relations ("who did what to whom?")
  - Good source of syntactic features for other tasks
- Easy to annotate (high agreement)
- Easy to evaluate (what % of words have correct parent?)

Supervised Learning Methods

- Easy to build a "yes" or "no" predictor from supervised training data
  - Plenty of software packages to do the learning & prediction
  - Lots of people in NLP never go beyond this ☺
- Similarly, easy to build a system that chooses from a small finite set
  - Basically the same deal
- Harder to predict the best string or tree (set is exponentially large or infinite)
  - Requires dynamic programming; you might have to write your own code
  - But finite-state or CRF toolkits will find the best string for you
  - And you could modify someone else's parser to pick the best tree
- Hardest if your features look at "non-local" properties of the string or tree
  - Now dynamic programming won't work (or will be something awful like $O(n^9)$)
  - You need some kind of approximate search
  - Can be harder to turn approximate search into a learning algorithm
  - Still, this is a standard preoccupation of machine learning ("structured prediction," "graphical models")
  - Using neural networks for learning transitions might help, but be careful with WSJ.

Text Annotation Tasks

1. Classify the entire document
2. Classify individual word tokens
3. Identify phrases (“chunking”)
4. Syntactic annotation (parsing)
5. Semantic annotation

Semantic Role Labeling (SRL)

- For each predicate (e.g., verb)
  1. find its arguments (e.g., NPs)
  2. determine their **semantic roles**

John drove Mary from Austin to Dallas in his Toyota Prius.

The hammer broke the window.

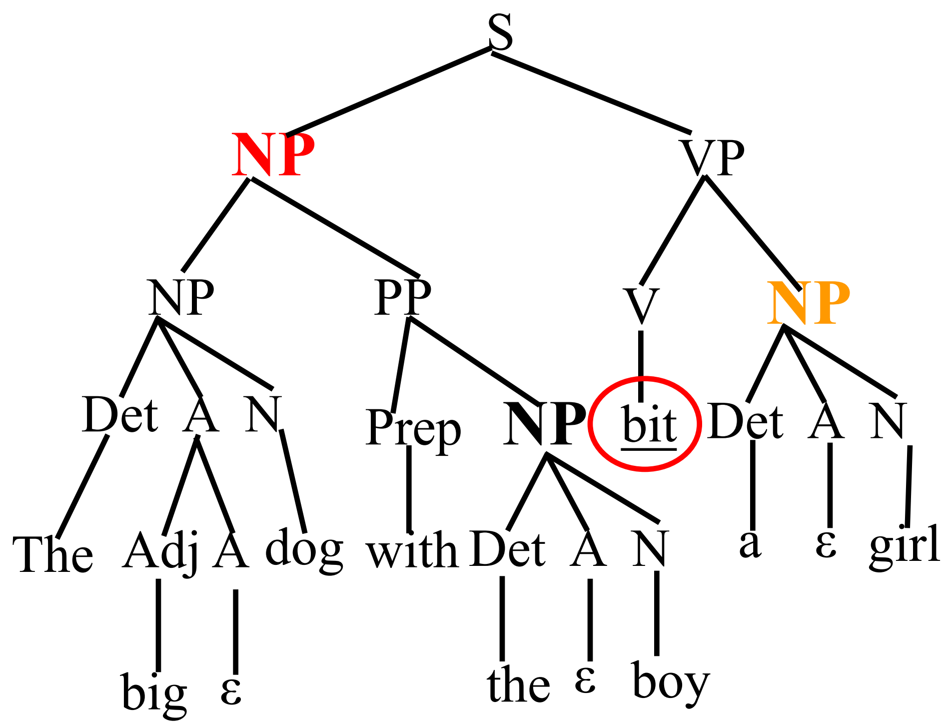
- agent: Actor of an action
- patient: Entity affected by the action
- source: Origin of the affected entity
- destination: Destination of the affected entity
- instrument: Tool used in performing action.
- beneficiary: Entity for whom action is performed

Slide thanks to Ray Mooney (modified)

As usual, can solve as classification …

- Consider one verb at a time: "bit"
- Classify the role (if any) of each of the 3 NPs

Color Code:
not-a-role
agent
patient
source
destination
instrument
beneficiary

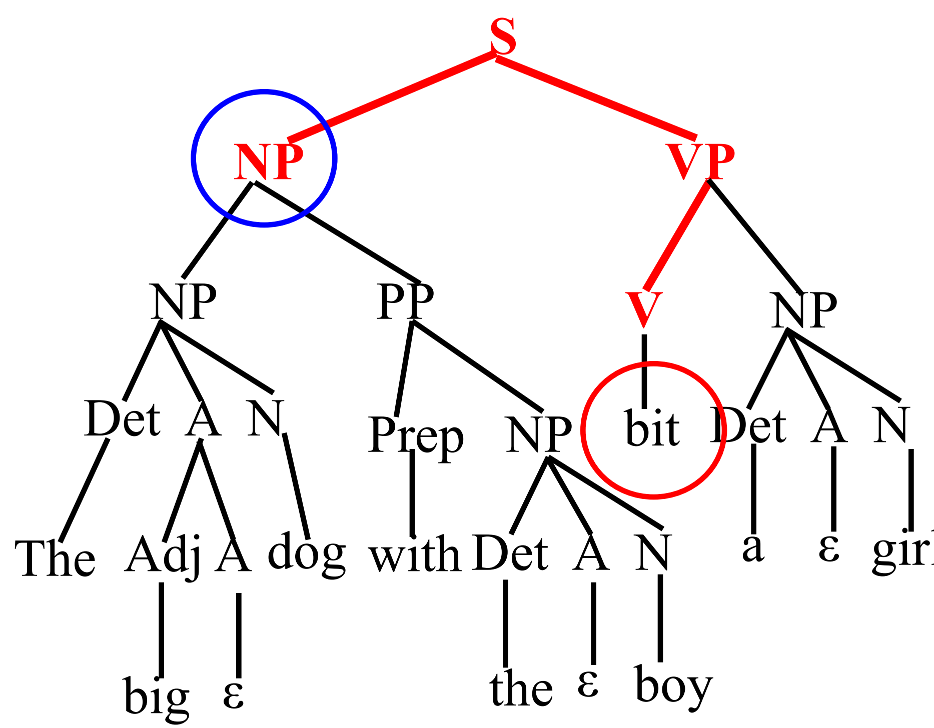


Parse tree paths as classification features

Path feature is

V ↑ VP ↑ S ↓ NP

which tends to be associated with agent role



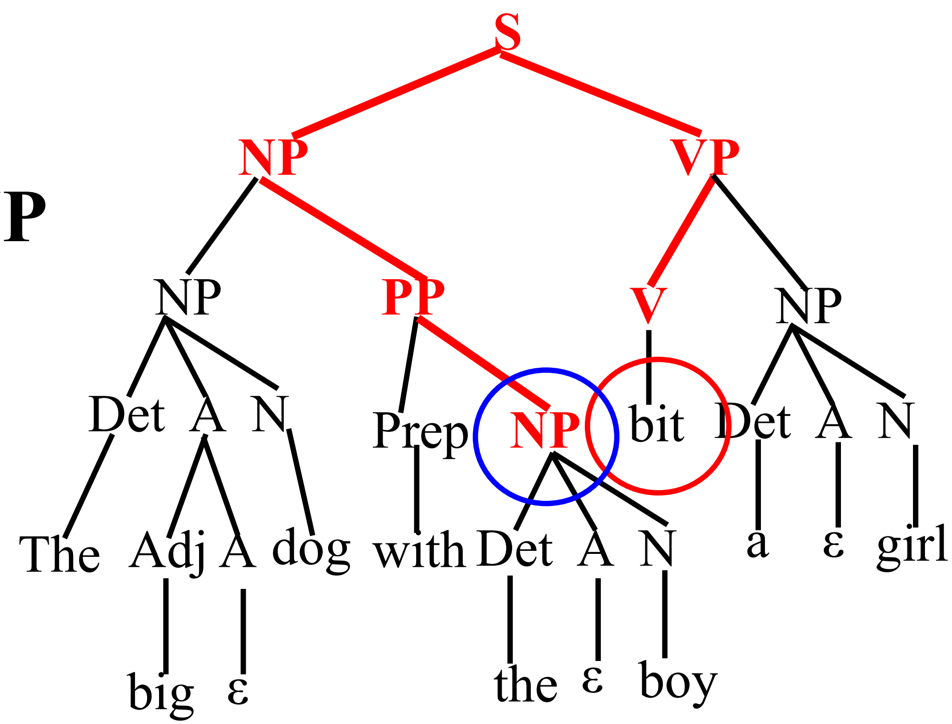
Slide thanks to Ray Mooney (modified)

Parse tree paths as classification features

Path feature is

V ↑ VP ↑ S ↓ NP ↓ PP ↓ NP

which tends to be associated with no role



Head words as features

- Some roles prefer to be filled by certain kinds of NPs.
- This can give us useful features for classifying accurately:
  - "John ate the spaghetti with chopsticks." **(instrument)**
    "John ate the spaghetti with meatballs." **(patient)**
    "John ate the spaghetti with Mary."
    - Instruments should be tools
    - Patient of "eat" should be edible
  - "John bought the car for $21K." **(instrument)**
    "John bought the car for Mary." **(beneficiary)**
    - Instrument of "buy" should be Money
    - Beneficiaries should be animate (things with desires)
  - "John drove Mary to school in the van"
    "John drove the van to work with Mary."
    - What do you think?

Slide thanks to Ray Mooney (modified)

Uses of Semantic Roles

- Find the answer to a user's question
  - "Who" questions usually want Agents
  - "What" question usually want Patients
  - "How" and "with what" questions usually want Instruments
  - "Where" questions frequently want Sources/Destinations.
  - "For whom" questions usually want Beneficiaries
  - "To whom" questions usually want Destinations
- Generate text
  - Many languages have specific syntactic constructions that must or should be used for specific semantic roles.
- Word sense disambiguation, using selectional restrictions
  - The **bat** ate the **bug**. (what kind of bat? what kind of bug?)
    - Agents (particularly of "eat") should be animate – animal bat, not baseball bat
    - Patients of "eat" should be edible – animal bug, not software bug
  - John **fired** the secretary.

    John **fired** the rifle.

    Patients of fire_1 are different than patients of fire_2

Slide thanks to Ray Mooney (modified)

Other Current Semantic Annotation Tasks (similar to SRL)

- PropBank – coarse-grained roles of verbs
- NomBank – similar, but for nouns
- FrameNet – fine-grained roles of any word
- TimeBank – temporal expressions

FrameNet Example

REVENGE FRAME

Avenger
Offender (unexpressed in this sentence)
Injury
Injured Party (unexpressed in this sentence)
Punishment

We avenged the insult by setting fire to his village.

a word/phrase that triggers the REVENGE frame

Slide thanks to CJ Fillmore (modified)

FrameNet Example

REVENGE FRAME
triggering words and phrases
(not limited to verbs)

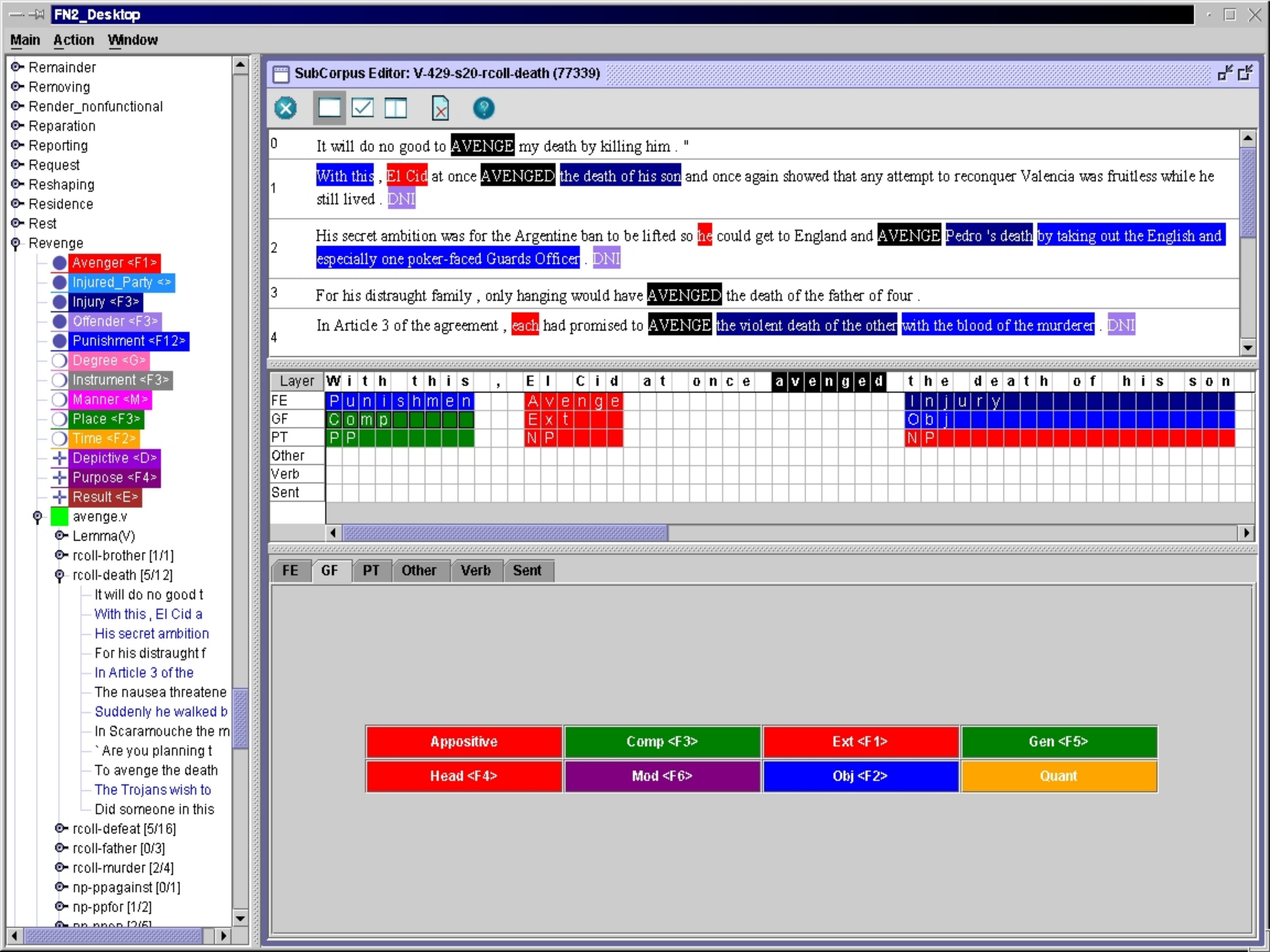
avenge, revenge, retaliate, get back at, pay back, get even, …

revenge, vengeance, retaliation, retribution, reprisal, …

vengeful, retaliatory, retributive; in revenge, in retaliation, …

take revenge, wreak vengeance, exact retribution, …

Slide thanks to CJ Fillmore (modified)



FN2_Desktop
Main Action Window
Remainder
Removing
Render_nonfunctional
Reparation
Reporting
Request
Reshaping
Residence
Rest
Revenge
Avenger <F1>
Injured_Party <>
Injury <F3>
Offender <F3>
Punishment <F12>
Degree <G>
Instrument <F3>
Manner <M>
Place <F3>
Time <F2>
Depictive <D>
Purpose <F4>
Result <E>
avenge.v
Lemma(V)
rcoll-brother [1/1]
rcoll-death [5/12]
It will do no good t
With this , El Cid a
His secret ambition
For his distraught f
In Article 3 of the
The nausea threatene
Suddenly he walked b
In Scaramouche the m
` Are you planning t
To avenge the death
The Trojans wish to
Did someone in this
rcoll-defeat [5/16]
rcoll-father [0/3]
rcoll-murder [2/4]
np-ppagainst [0/1]
np-ppfor [1/2]
SubCorpus Editor: V-429-s20-rcoll-death (77339)
0 It will do no good to AVENGE my death by killing him . "
1 With this , El Cid at once AVENGED the death of his son and once again showed that any attempt to reconquer Valencia was fruitless while he still lived . DNI
2 His secret ambition was for the Argentine ban to be lifted so he could get to England and AVENGE Pedro 's death by taking out the English and especially one poker-faced Guards Officer . DNI
3 For his distraught family , only hanging would have AVENGED the death of the father of four .
4 In Article 3 of the agreement , each had promised to AVENGE the violent death of the other with the blood of the murderer . DNI
Layer
With this , El Cid at once avenged the death of his son
FE Punishmen Avenge Injury
GF Comp Ext Obj
PT PP NP NP
Other
Verb
Sent
FE GF PT Other Verb Sent
Appositive
Comp <F3>
Ext <F1>
Gen <F5>
Head <F4>
Mod <F6>
Obj <F2>
Quant

Generating new text

1. Speech recognition (transcribe as text)
2. Machine translation
3. Text generation from semantics
4. Inflect, analyze, or transliterate words
5. Single- or multi-doc summarization

Deeper Information Extraction

1. Coreference resolution (within a document)
2. Entity linking (across documents)
3. Event extraction and linking
4. Knowledge base population (KBP)
5. Recognizing texual entailment (RTE)

User interfaces

1. Dialogue systems
   - Personal assistance
   - Human-computer collaboration
   - Interactive teaching
2. Language teaching; writing help
3. Question answering
4. Information retrieval

Multimodal interfaces or modeling

1. Sign languages
2. Speech + gestures
3. Images + captions
4. Brain recordings, human reaction times

NLP automates things that humans do well, so that they can be done automatically on more sentences. But this slide is about language analysis that's hard even for humans. Computational linguistics (like comp bio, etc.) can discover underlying patterns in large datasets: things we didn't know!

Discovering Linguistic Structure

1. Decipherment
2. Grammar induction
3. Topic modeling
4. Deep learning of word meanings
5. Language evolution (historical linguistics)
6. Grounded semantics