Syntactic Analysis: POS Tagging

Natural Language Processing

Part-of-Speech (POS) Tags

In WordNet we saw a very basic set of POS tags:

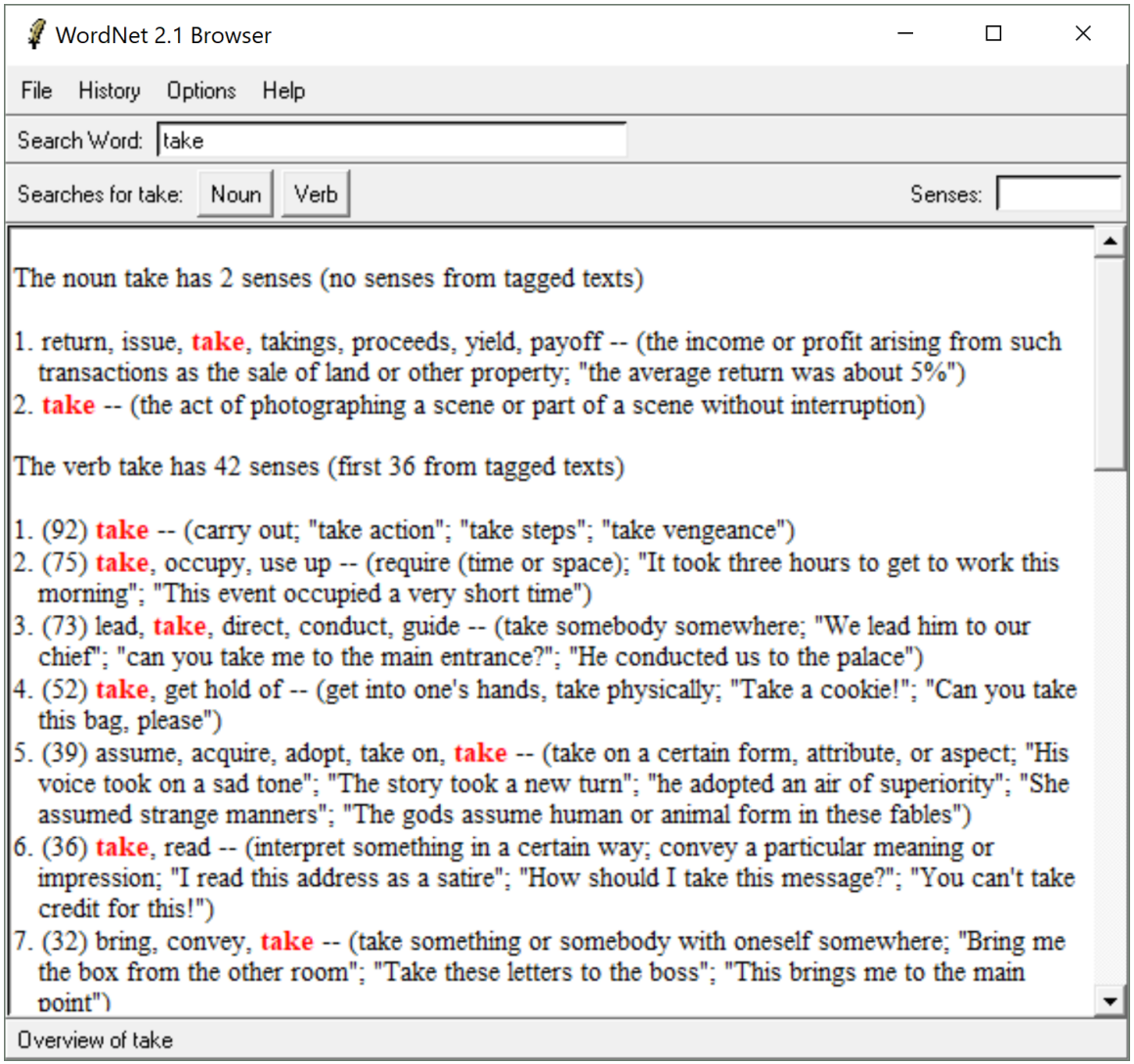
- N = noun
- V = verb
- A = adjective
- R = adverb

The lexical knowledge base WordNet was made by Princeton University and is widely used in NLP.

POS Tags

We saw that we can get a much better representation of appropriate word senses if we know which POS is intended.

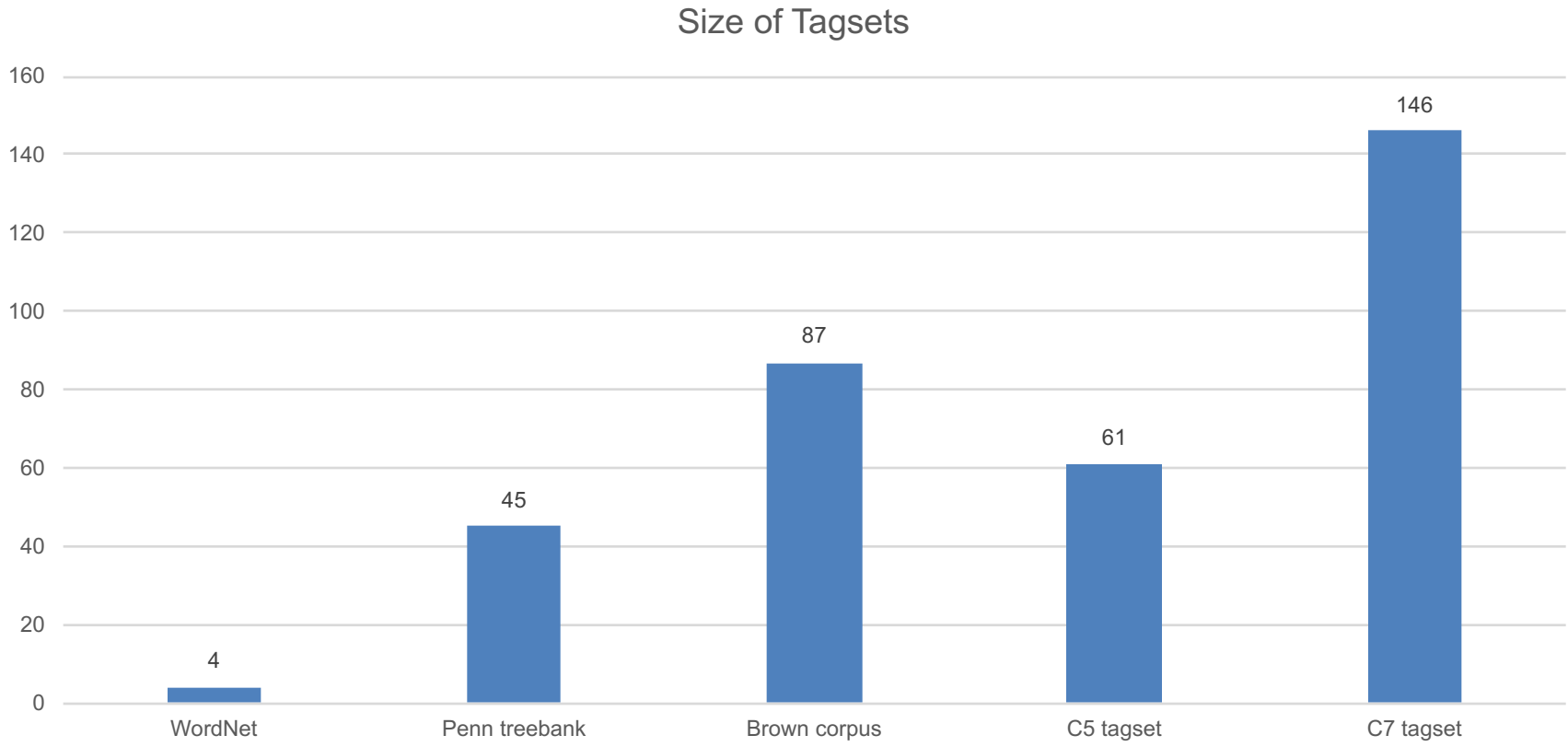
Consider how different the noun senses are from the verb senses for “take”…



Penn Treebank Tagset

As the most widely used tagset, this has *way* more than 4 tags—45 to be exact.

Other tagsets are excessively detailed (beyond what's needed for most of our applications).



Penn Treebank Tagset

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*			

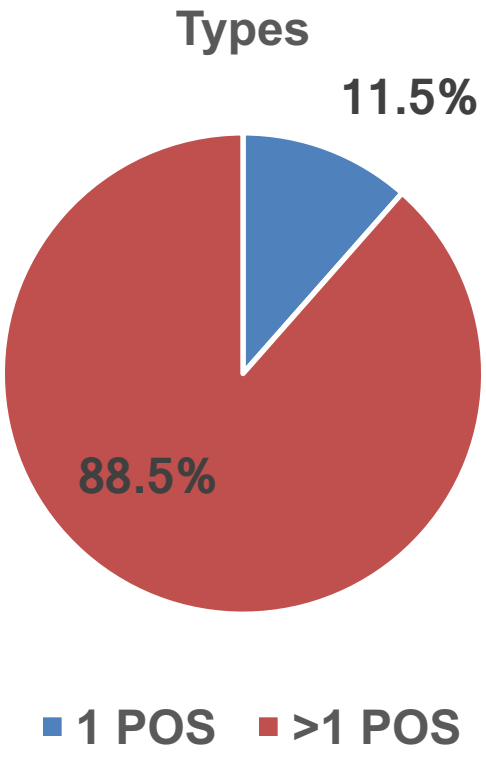
Penn Treebank Tagset

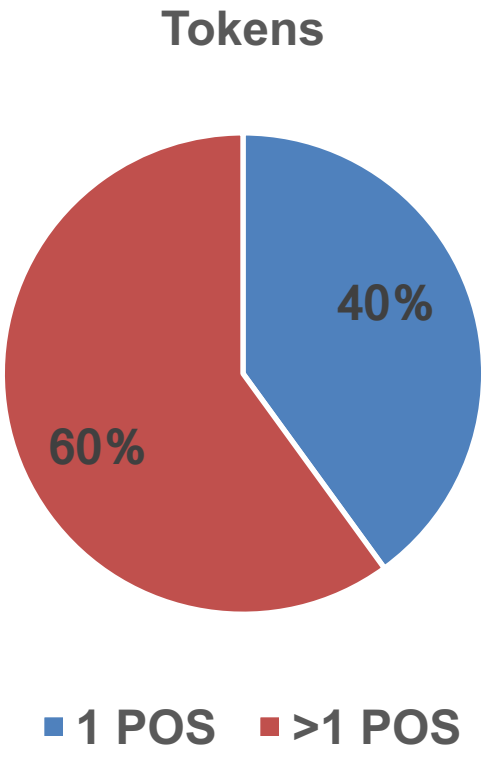
Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	+,%, &
CD	Cardinal number	*one, two, three*	TO	"to"	*to*
DT	Determiner	*a, the*	UH	Interjection	*ah, oops*
EX	Existential 'there'	*there*	VB	Verb, base form	*eat*
FW	Foreign word	*mea culpa*	VBD	Verb, past 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*			

Susan decided to run an errand while Joe went on his run.
NNP VBD TO VB DT NN IN NNP VBD IN PRP$ NN

How Ambiguous Is POS?

Based on the Brown corpus:





DataScience@SMU

Syntactic Analysis:
POS Tagging—How POS Taggers Work

Natural Language Processing

How POS Taggers Work: A Rule-Based POS Tagger

Eric Brill, 1993

Two basic steps:

1. Assign most common POS tag initially (knowing there will be a lot of errors).
2. Check the rule-base for transformation rules, e.g., rules that correct errors based on words or tags.

```
The running of the bulls in Spain is very popular.
DT  VBG     IN DT  NNS   IN NNP   VBG RB  JJ

    NN
↓
Rule: VBG → NN if followed by "of"
```

Rules could be handcrafted or generated by supervised learning or a combination.

How POS Taggers Work

Generally there is a sliding window.

How POS Taggers Work

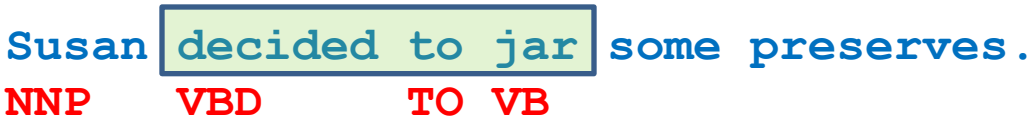
Generally there is a sliding window.

How POS Taggers Work

Generally there is a sliding window.

How POS Taggers Work

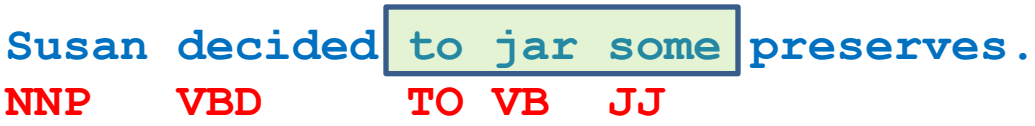
Generally there is a sliding window.



Notice the adjustment that just happened?

How POS Taggers Work

Generally there is a sliding window.



How POS Taggers Work

Generally there is a sliding window.

Susan	decided	to	jar	some	preserves.
NNP	VBD	TO	VB	JJ	NNS

How POS Taggers Work

Generally there is a sliding window.

How POS Taggers Work

Generally there is a sliding window.

How POS Taggers Work

And we can go backwards, too…

How POS Taggers Work

Backward-sliding window:

How POS Taggers Work

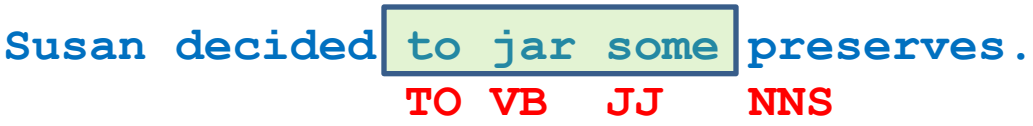
Backward-sliding window:

How POS Taggers Work

Backward-sliding window:

How POS Taggers Work

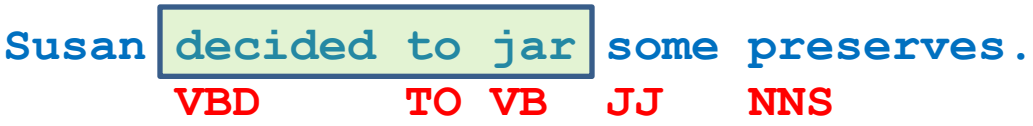
Backward-sliding window:



Notice the adjustment that just happened?

How POS Taggers Work

Backward-sliding window:



How POS Taggers Work

Backward-sliding window:

How POS Taggers Work

Backward-sliding window:

How POS Taggers Work

Backward-sliding window:

DataScience@SMU

Syntactic Analysis: POS Tagging (Continued)

Natural Language Processing

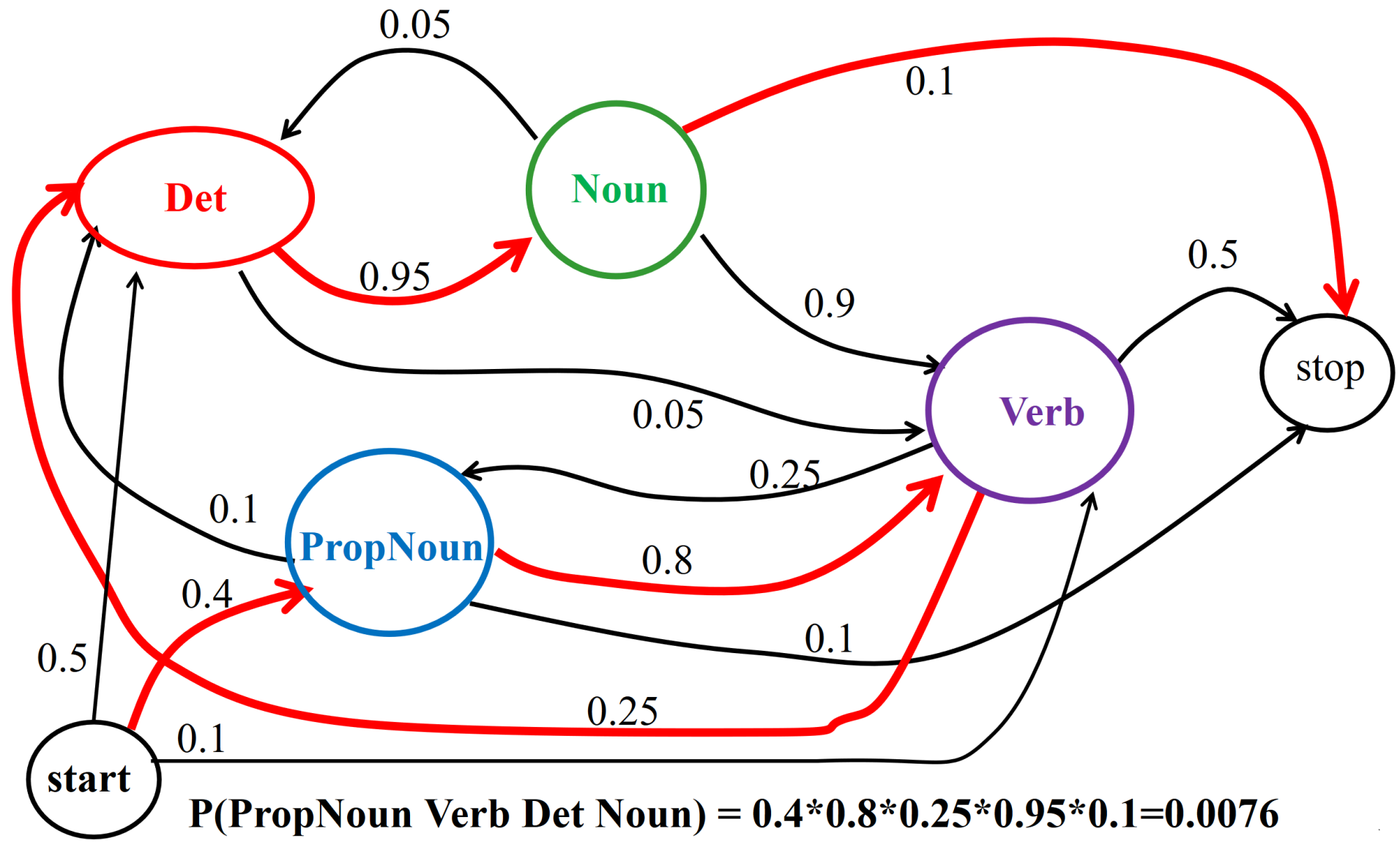
An Early Approach to Statistical POS Tagging

- Hidden Markov models (HMMs) for POS tagging started in the 1980s.
- In simplified form, consider making a table of POS probabilities based on preceding word(s).

Tokens preceded by "the"	
Noun	51%
Adjective	29%
Number	16%
Adverb	4%

- After consulting all applicable tables for a token, select tag with the highest overall likelihood.

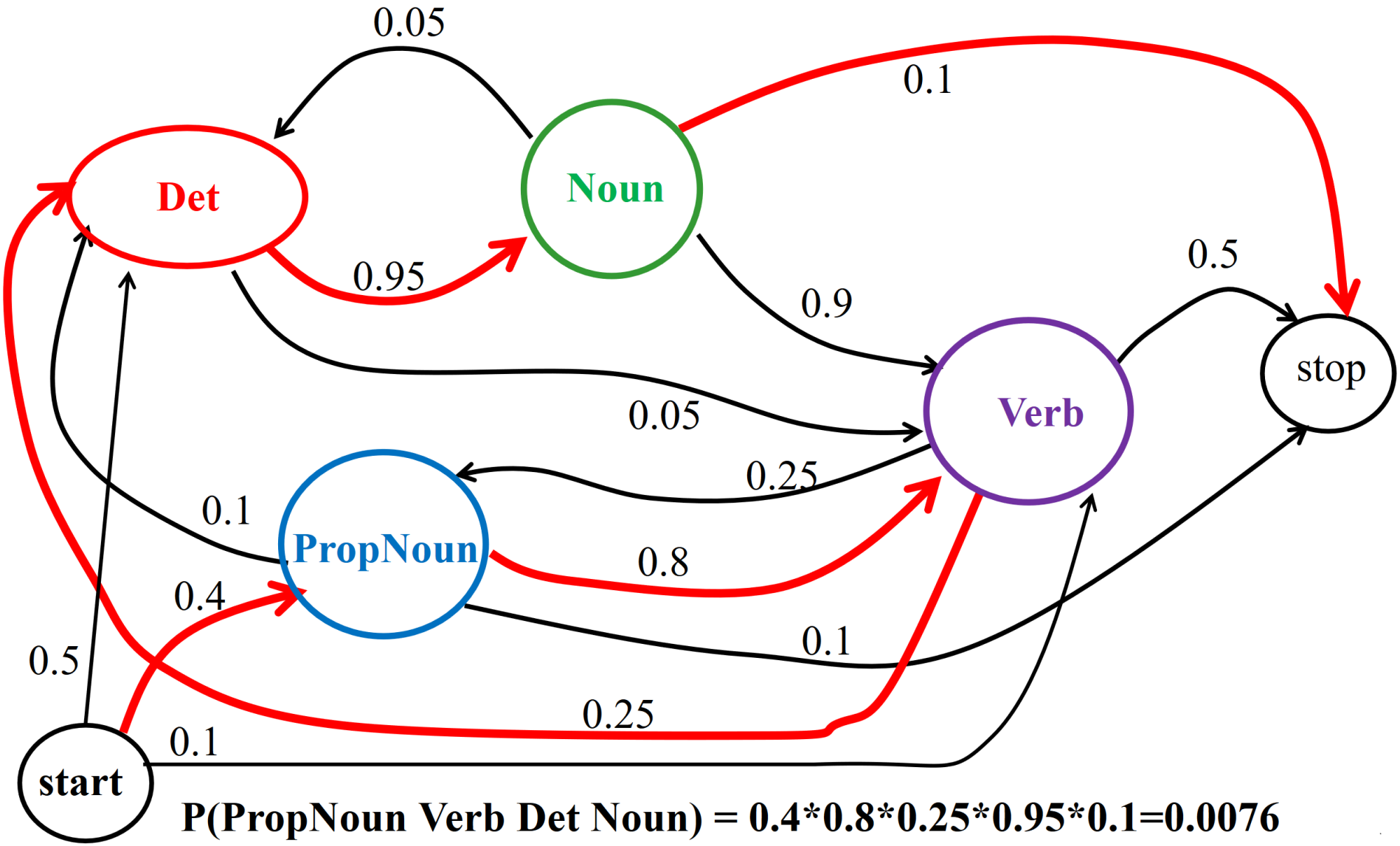
Example HMM for POS: “Mary saw a cat”



P(PropNoun Verb Det Noun) = 0.4*0.8*0.25*0.95*0.1=0.0076

Consider that “saw” could be a noun.
Why does it not get labelled a noun here?

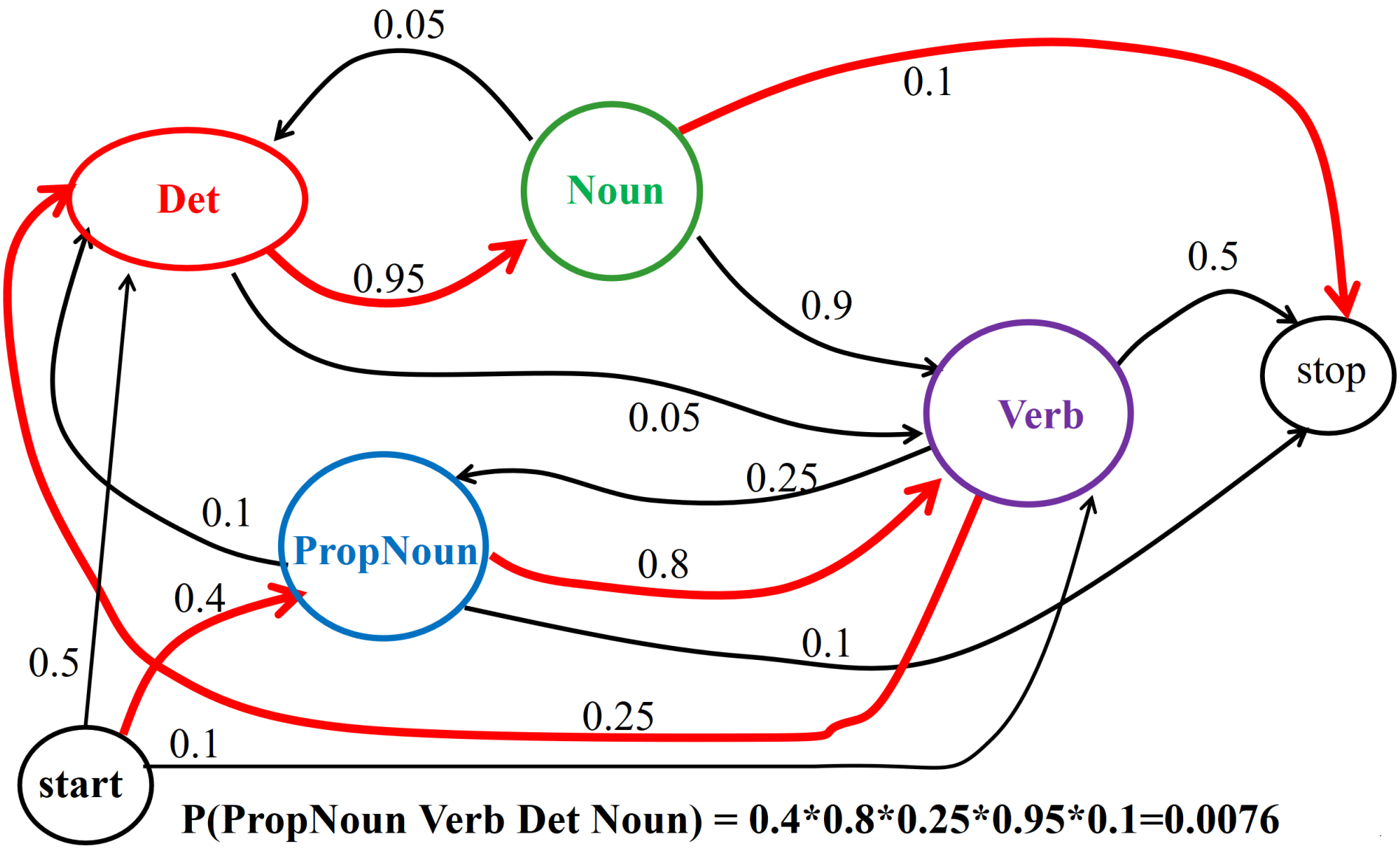
“Mary saw a cat”



P(PropNoun Verb Det Noun) = 0.4*0.8*0.25*0.95*0.1=0.0076

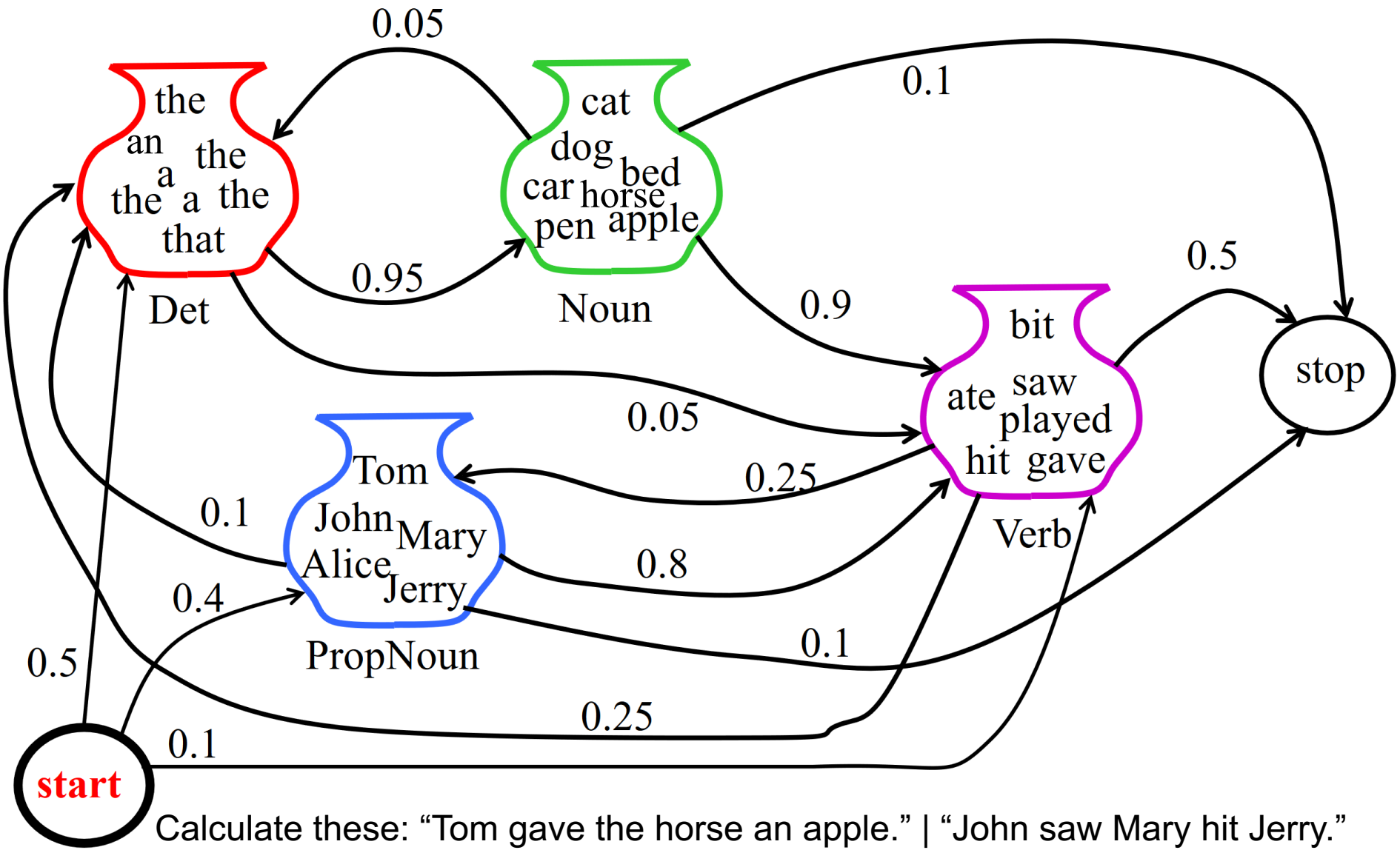
Consider that “cat” can actually be a verb. Why does it not get labelled a verb here?

“Mary saw a cat”



P(PropNoun Verb Det Noun) = 0.4*0.8*0.25*0.95*0.1=0.0076

Example HMM for POS



Calculate these: “Tom gave the horse an apple.” | “John saw Mary hit Jerry.”

Implementing HMM POS Tagging

- A useful HMM-based POS tagger needs so many HMMs that it's not practical to construct them manually.
- There's an algorithm for training, i.e., generating HMMs from labeled data (the *Baum-Welch algorithm).*
- There's an algorithm for examining a set of HMMs to determine the most likely sequence of "states" of the HMM, i.e., most likely sequence of POS tags (the *Viterbi algorithm*).

Andrew Viterbi

Various Approaches

- Rule based (usually handcrafted rules)
- Statistical (or stochastic) models
  - Hidden Markov model (HMM)
  - Maximum entropy Markov model (MEMM)
  - Conditional random field (CRF)
- Machine learning
  - N-gram trainer
  - Naïve Bayes
  - Neural network
  - Support Vector Machine (SVM)
  - Many of these are included in NLTK (natural language toolkit)

Automated POS Tagging

- Typically, we use ML (machine learning) on tagged training sets to build automatic POS taggers.
- Here's an example using a Naïve Bayes classifier-based tagger, trained on the Penn Treebank:

```
>>> sentence
'The rain in Spain stays mainly on the plain'
>>> tokens
['The', 'rain', 'in', 'Spain', 'stays', 'mainly', 'on', 'the', 'plain']
>>> nbt.tag(tokens)
[('The', 'DT'), ('rain', 'NN'), ('in', 'IN'), ('Spain', 'NNP'), ('stays',
 'VBZ'), ('mainly', 'RB'), ('on', 'IN'), ('the', 'DT'), ('plain', 'NN')]
>>>
```

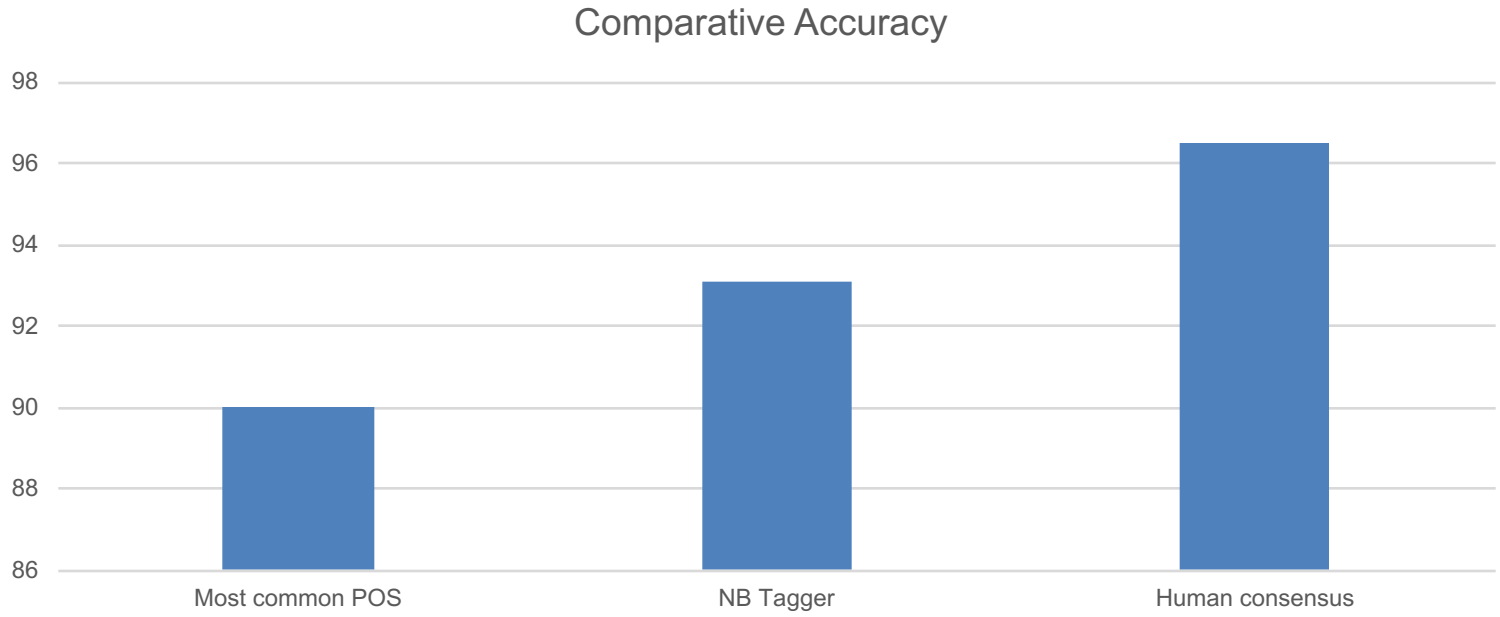
Training Data

- First we need training data: the Penn Treebank will do.
- It has over 7 million words of manually POS-tagged text, created at the University of Pennsylvania between 1989 and 1996 and covering a broad range of material including:
  - IBM manuals
  - Nursing notes
  - *Wall Street Journal* articles
  - Transcribed telephone conversations
  - Much more

How Good Is It?

Using the Penn Treebank for both training and test data, we yield about 93.1% accuracy with the Naïve Bayes tool built into NLTK. The best taggers in use today achieve around 96% accuracy.

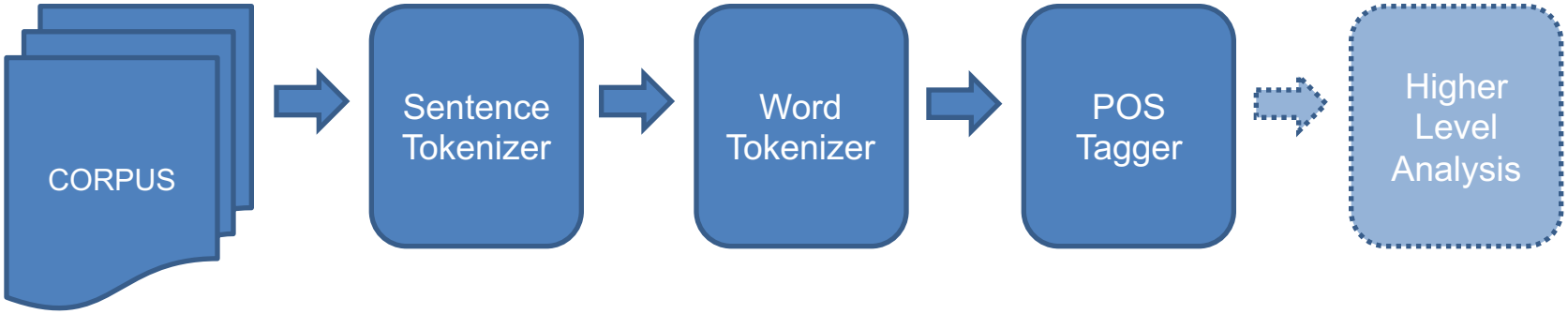
- In the Brown corpus, just guessing the most common POS for each word gives 90% accuracy.
- Human annotators for the Penn Treebank disagreed with each other in 3.5% of cases—implying that attempting better than 96.5% accuracy in POS tagging might not be particularly meaningful.



How Do We Get There?

Procedurally, tokenized text can be fed into a trained POS tagger.

- And you can experiment with many other POS taggers, singly or in combination.



Pipeline processing

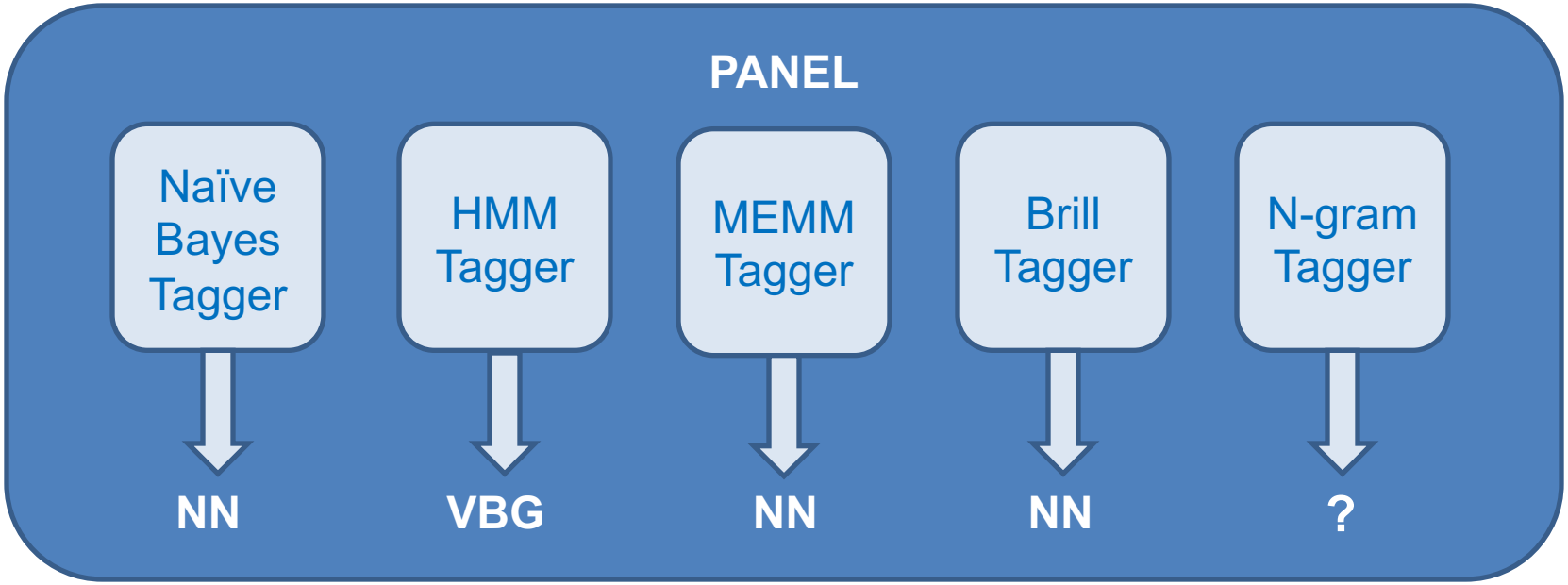
Combining Taggers

Bonus question:

How would you systematically combine three or more POS taggers?

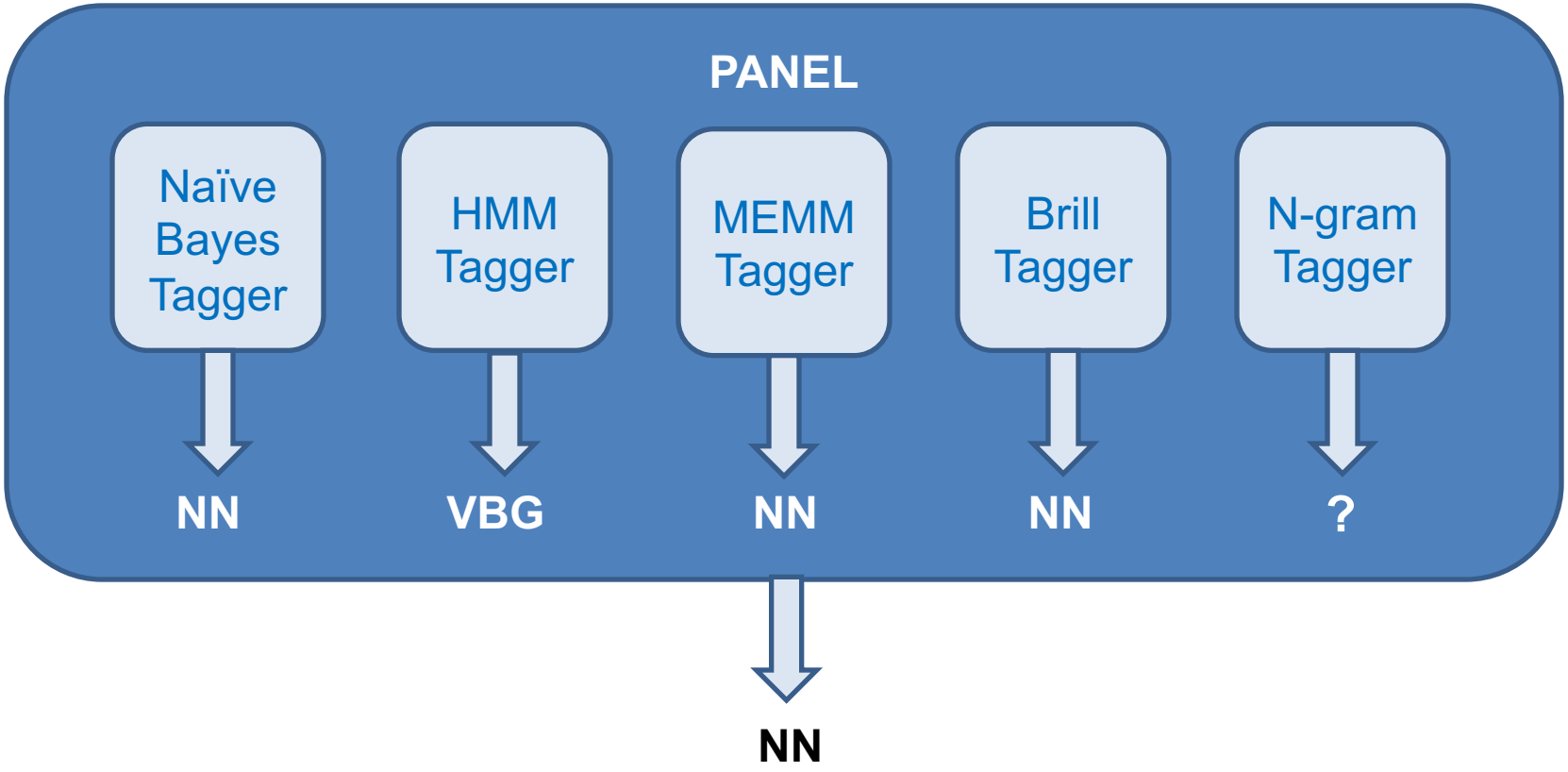
Combining Taggers

A voting engine can be like a “Supreme Court”—majority rules!



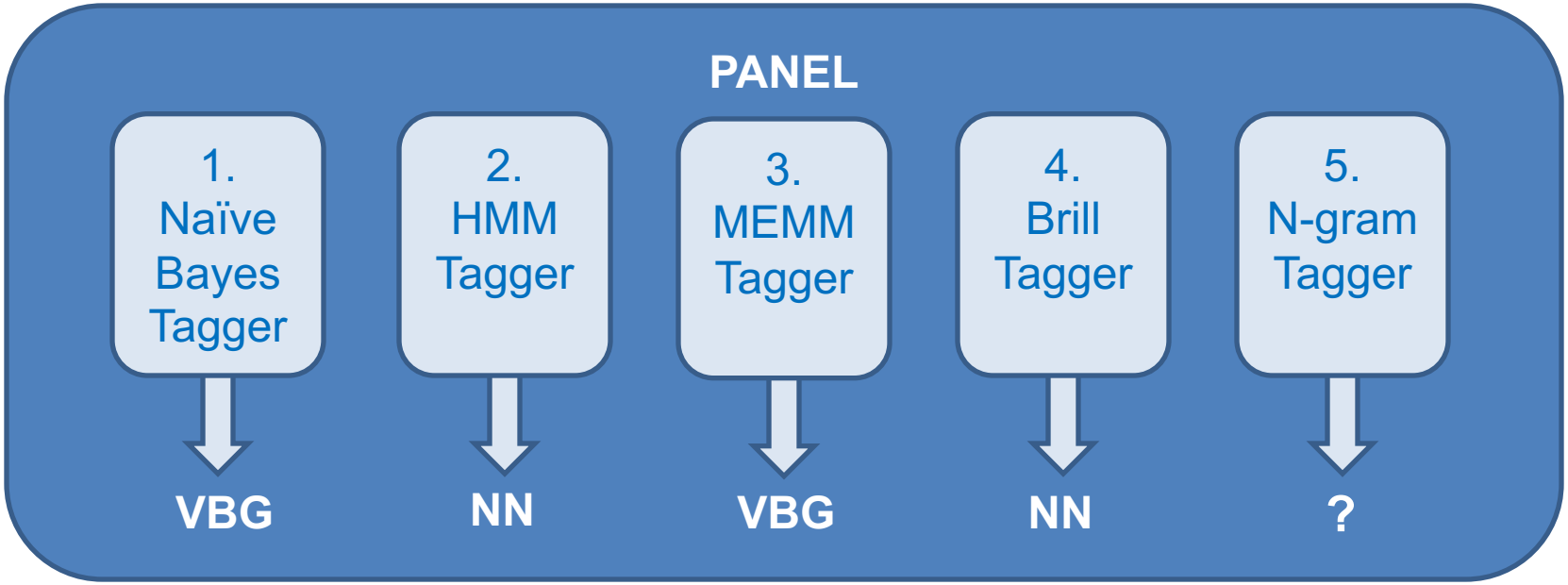
Combining Taggers

A voting engine can be like a “Supreme Court”—majority rules!



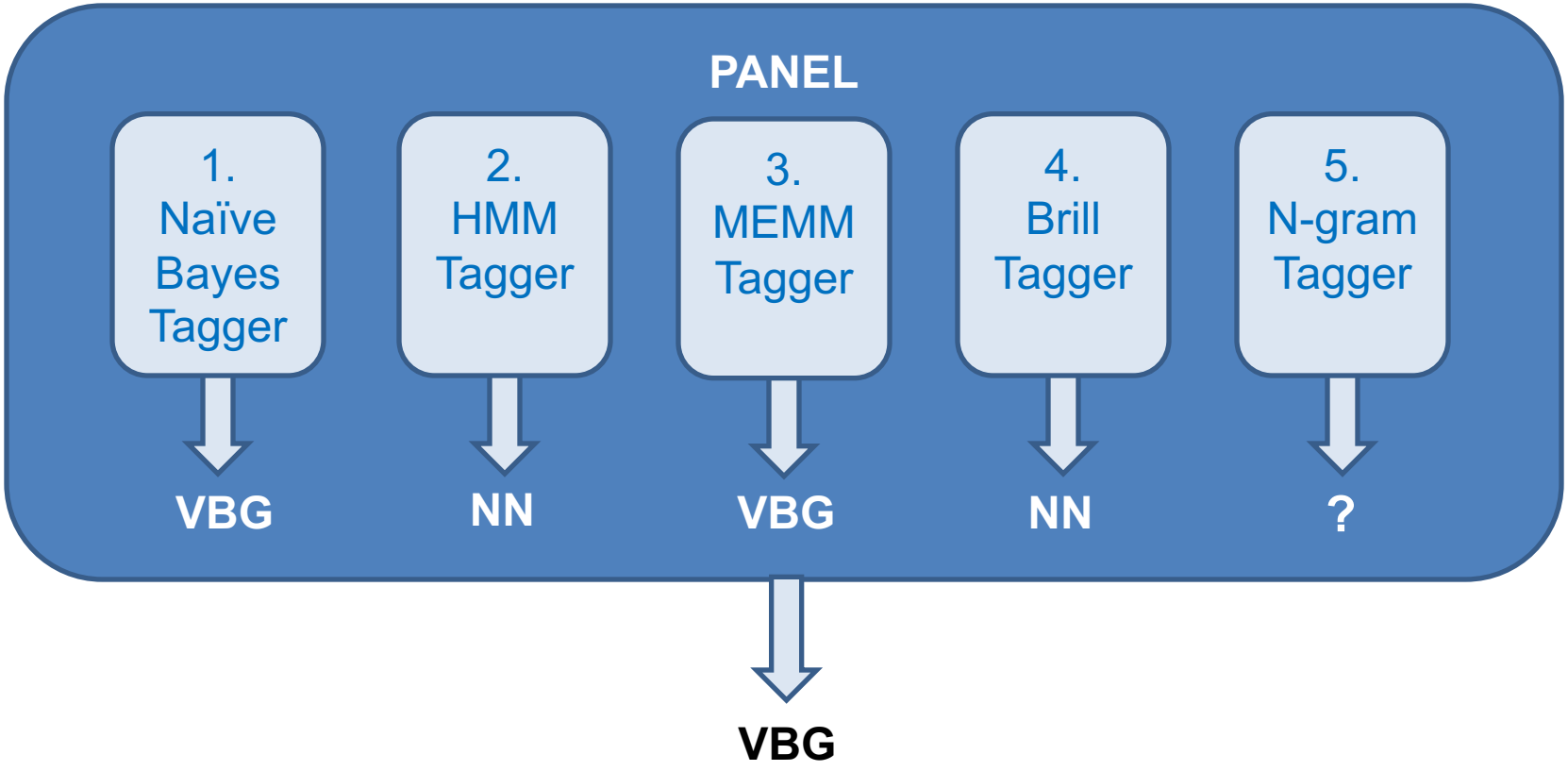
Combining Taggers

Tie-break example:



Combining Taggers

Tie-break example:



DataScience@SMU

Syntactic Analysis: Using Part-of-Speech Tags

Natural Language Processing

Using POS Tags

Simple uses for POS tags

```
Susan decided to run an errand while Joe went on his run.
PPN   VBD     TO VB  DT NN     IN    NNP VBD  IN PRP$ NN
```

- Starting to narrow down synset look-ups in WordNet
- Rudimentary NER
- Potentially more useful word clouds
- More robust sentiment analysis

Using POS Tags with WordNet

We want to narrow down which synsets might be relevant to an instance of a word in a document.

```
Leon said the officer was on the take.
NNP  VBD  DT  NN      VBD IN DT  NN
```

Consider the word “take” above.

Using POS Tags with WordNet

```
Leon said the officer was on the take.
NNP  VBD  DT  NN      VBD IN DT  NN
```

Disregarding POS, “take” has 44 senses in WordNet, but as a noun, just two:

1. *return, issue, take, takings, proceeds, yield, payoff—(the income or profit arising from such transactions as the sale of land or other property; "the average return was about 5%")*
2. *take—(the act of photographing a scene or part of a scene without interruption)*

Admittedly, this doesn’t get us all the way to automated WSD (nothing does!), but it’s a big step.

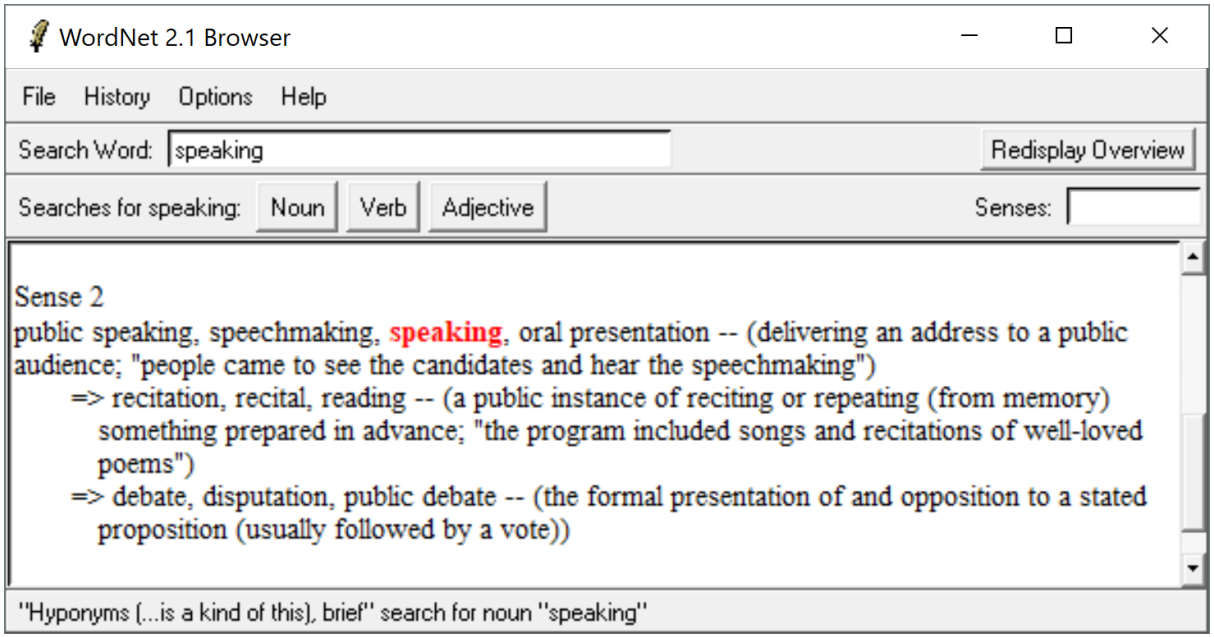
Using POS Tags with WordNet

Knowing the likely POS for a keyword in a search query helps us when considering synonyms/hyponyms to substitute as related search terms.

Query: ***lessons on speaking***

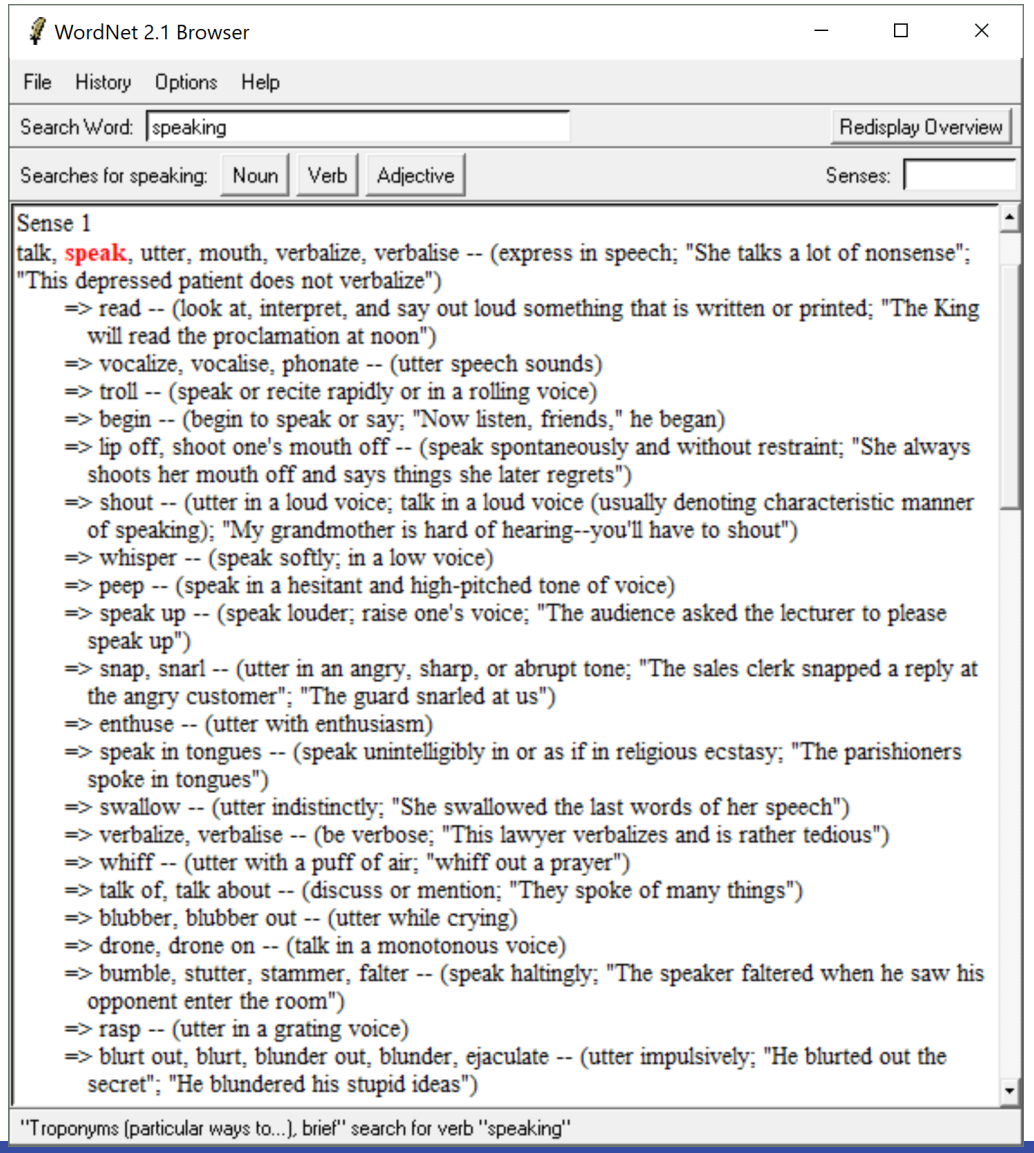
See also:
lessons on public debate

Knowing that "speaking" is tagged as a noun, we take only noun hyponyms (filtered via pseudo-relevance feedback).



Using POS Tags with WordNet

Just look at all the verb troponyms we spared ourselves from considering.



Using POS Tags with WordNet

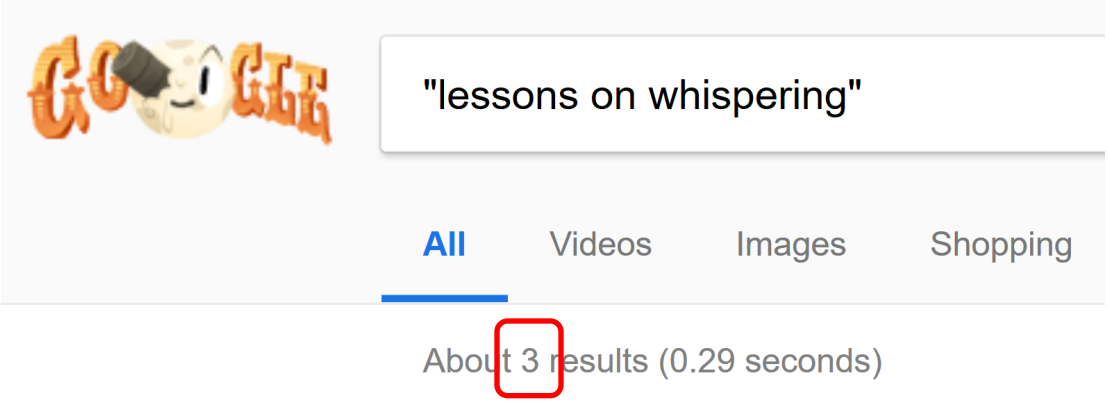
- But wait—“whispering” is listed in WordNet as a hyponym of N1 of “speaking”!
- So what saves us from creating this?

Query: ***lessons on speaking***

See also:
lessons on whispering

Using POS Tags with WordNet

Here's what saves us:



vs.

DataScience@SMU

Syntactic Analysis: Using Part-of-Speech Tags

Natural Language Processing

Using POS Tags in NER

We could construe any run of NNPs as a NE.

Susan	Mathison	gave	the	letter	to	Joseph	Baxter.
NNP	NNP	VBD	DT	NN	TO	NNP	NNP

And break out separate NEs when we see intervening punctuation.

While	Bob	talked	to	Carlos	,	Susan	talked	to	Ayla.
PP	NNP	VBD	TO	NNP	,	NNP	VBD	TO	NNP

Using POS Tags in NER

But that method is prone to occasional error…

```
Susan told Alice Raul was at home.
NNP   VBD  NNP   NNP  VBD PP NN
```

…and problems come up with punctuation.

```
The letter came from the law firm of Muir, Wilson & Chen.
```

Despite these occasional pitfalls, this method would still be good enough for some applications.

Using POS Tags in NER

- The NNP-collocations method, simple as it is, would give us a list of *candidate* NEs.
- In many applications, we want to output only NEs that are already *designated* in some knowledge base, e.g., Wikipedia, Standard & Poor's, IMDB.

STANDARD &POOR'S

IMDb

Using POS Tags in NER

- By looking up candidate NEs in such a knowledge base, we would weed out many of the errant NE candidate strings.
- This would eliminate “noise” (erroneous candidates), making the output usable for many applications, e.g., autogenerating an index page for designated NEs that are mentioned in a corpus.

NNP-string	Look-up
Susan Mathison	Found
Joseph Baxter	Found
Alice Raul	Not Found
Chen	Not Found

Using POS Tags in NER

- Another way of eliminating NE candidates is just looking up their document frequency in the corpus.
- This would eliminate (and keep) different NNP-strings.

NNP-string	Frq
Susan Mathison	37
Joseph Baxter	3
Alice Raul	1
Chen	48

Later we'll look at dedicated NER engines that can do an even *better* job for us.

Conjunctive Elimination

- We can combine both methods of candidate validation/elimination.
- We eliminate a candidate NNP-collocation IF-AND-ONLY-IF it is not found in the NE database AND it has low frequency.

NNP-string	Look-up
Susan Mathison	Found
Joseph Baxter	Found
~~Alice Raul~~	Not Found
Chen	Not Found

NNP-string	Frq
Susan Mathison	37
Joseph Baxter	3
~~Alice Raul~~	1
Chen	48

DataScience@SMU

Syntactic Analysis: Using Part-of-Speech Tags

Natural Language Processing

Using POS Tags in Word Clouds

Here's a POS-blind word cloud generated from a review of a music concert:

Using POS Tags in Word Clouds

Here's how it regenerates if we eliminate verbs:

Using POS Tags in Sentiment Analysis

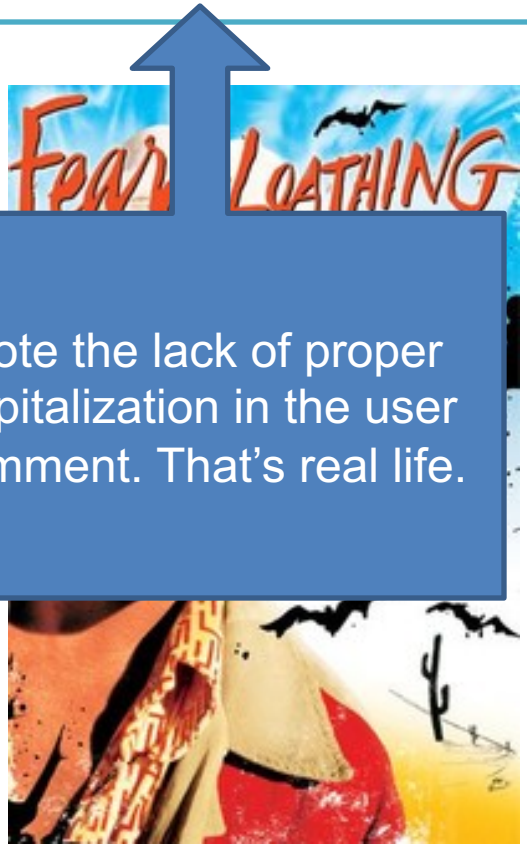
"I found this is an exciting movie, it's depp's most inspiring performance since fear and loathing in las vegas."

- Clearly (to us humans) this user comment has overall *positive* sentiment.
- But because of the negative words, it would likely register as overall *negative sentiment* to a basic, weighted-vocabulary sentiment analyzer.
- That's because a *stemmer* would be invoked to treat "loathe, loathed, loathing" all the same way.
- But with a POS tagger, you could designate that the gerund form of words like "loathe" don't count as negative, or not as *highly* negative, in your project.

Using POS Tags in Sentiment Analysis

"I found this is an exciting movie, it's depp's most inspiring performance since fear and loathing in las vegas."

- Clearly (to us humans) this user comment has overall *positive* sentiment.
- But because of the negative words, it would likely register as overall *negative sentiment* to a basic, weighted-vocabulary sentiment analyzer.
- That's because a *stemmer* would be invoked to treat "loathe, loathed, loathing" all the same way.
- But with a POS tagger, you could designate that the gerund form of words like "loathe" don't count as negative, or not as *highly* negative, in your project.



Note the lack of proper capitalization in the user comment. That's real life.

DataScience@SMU