

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.

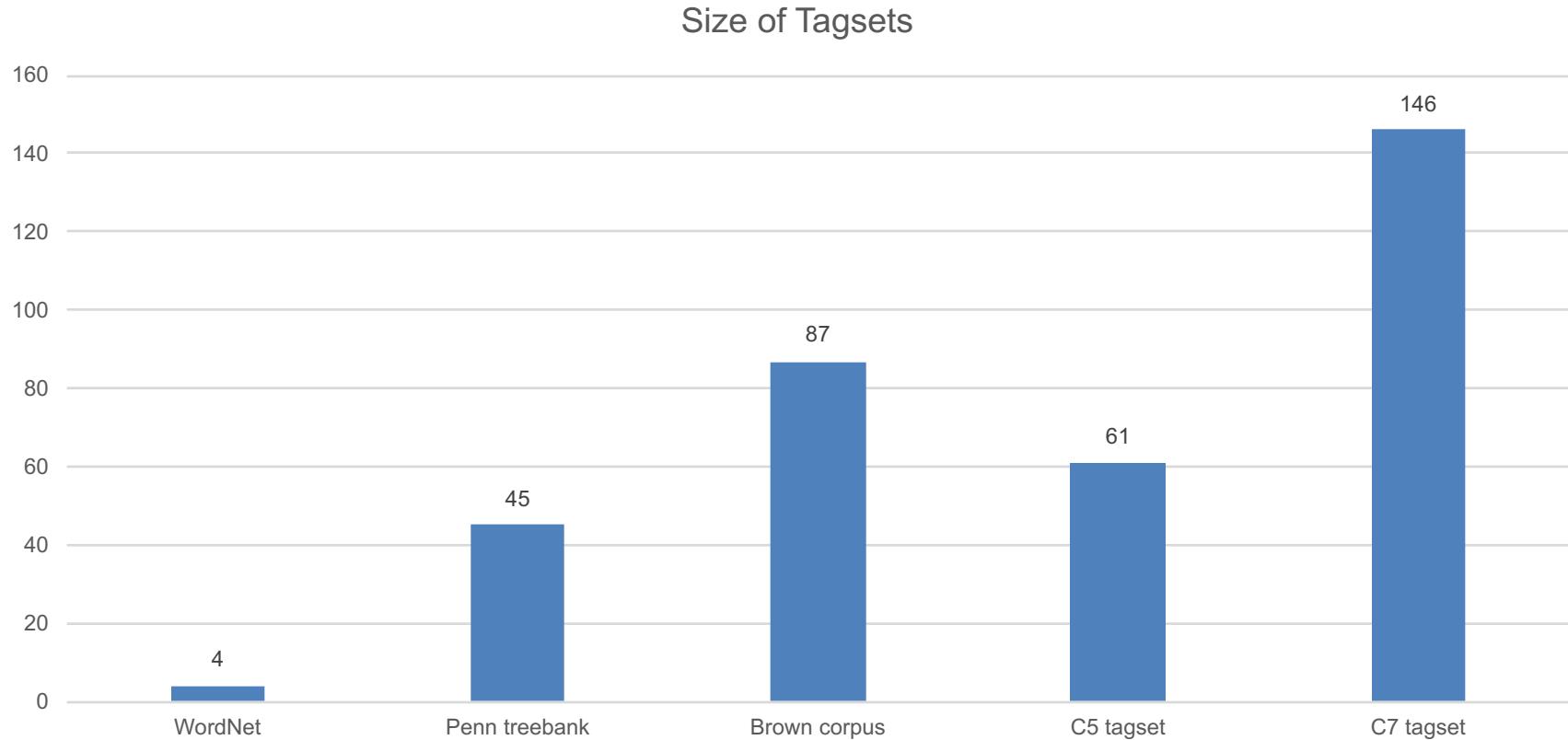
Consider how different the noun senses are from the verb senses for “take”...

The screenshot shows the WordNet 2.1 Browser interface. The title bar says "WordNet 2.1 Browser". The menu bar includes "File", "History", "Options", and "Help". The search bar contains the word "take". Below the search bar, there are buttons for "Searches for take: Noun" and "Verb". To the right of the search bar is a "Senses:" button. The main content area displays the results for the noun sense of "take". It states: "The noun take has 2 senses (no senses from tagged texts)". Below this, two numbered items list the noun senses: 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%")" and 2. "**take** -- (the act of photographing a scene or part of a scene without interruption)". The content then transitions to the verb sense of "take", stating: "The verb take has 42 senses (first 36 from tagged texts)". A numbered list follows, starting with: 1. "(92) **take** -- (carry out; "take action"; "take steps"; "take vengeance")", 2. "(75) **take**, occupy, use up -- (require (time or space); "It took three hours to get to work this morning"; "This event occupied a very short time")", 3. "(73) lead, **take**, direct, conduct, guide -- (take somebody somewhere; "We lead him to our chief"; "can you take me to the main entrance?"; "He conducted us to the palace")", 4. "(52) **take**, get hold of -- (get into one's hands, take physically; "Take a cookie!"; "Can you take this bag, please")", 5. "(39) assume, acquire, adopt, take on, **take** -- (take on a certain form, attribute, or aspect; "His voice took on a sad tone"; "The story took a new turn"; "he adopted an air of superiority"; "She assumed strange manners"; "The gods assume human or animal form in these fables")", 6. "(36) **take**, read -- (interpret something in a certain way; convey a particular meaning or impression; "I read this address as a satire"; "How should I take this message?"; "You can't take credit for this!")", and 7. "(32) bring, convey, **take** -- (take something or somebody with oneself somewhere; "Bring me the box from the other room"; "Take these letters to the boss"; "This brings me to the main point")". At the bottom of the content area is a link "Overview of 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	<i>and, but, or</i>	SYM	Symbol	+%, &
CD	Cardinal number	<i>one, two, three</i>	TO	"to"	<i>to</i>
DT	Determiner	<i>a, the</i>	UH	Interjection	<i>ah, oops</i>
EX	Existential 'there'	<i>there</i>	VB	Verb, base form	<i>eat</i>
FW	Foreign word	<i>mea culpa</i>	VBD	Verb, past tense	<i>ate</i>
IN	Preposition/sub-conj	<i>of, in, by</i>	VBG	Verb, gerund	<i>eating</i>
JJ	Adjective	<i>yellow</i>	VBN	Verb, past participle	<i>eaten</i>
JJR	Adj., comparative	<i>bigger</i>	VBP	Verb, non-3sg pres	<i>eat</i>
JJS	Adj., superlative	<i>wildest</i>	VBZ	Verb, 3sg pres	<i>eats</i>
LS	List item marker	<i>1, 2, One</i>	WDT	Wh-determiner	<i>which, that</i>
MD	Modal	<i>can, should</i>	WP	Wh-pronoun	<i>what, who</i>
NN	Noun, sing. or mass	<i>llama</i>	WP\$	Possessive wh-	<i>whose</i>
NNS	Noun, plural	<i>llamas</i>	WRB	Wh-adverb	<i>how, where</i>
NNP	Proper noun, singular	<i>IBM</i>	\$	Dollar sign	\$
NNPS	Proper noun, plural	<i>Carolinas</i>	#	Pound sign	#
PDT	Predeterminer	<i>all, both</i>	"	Left quote	(‘ or “)
POS	Possessive ending	<i>'s</i>	"	Right quote	(‘ or ”)
PRP	Personal pronoun	<i>I, you, he</i>	(Left parenthesis	([, (, {, <)
PRP\$	Possessive pronoun	<i>your, one's</i>)	Right parenthesis	(],), }, >)
RB	Adverb	<i>quickly, never</i>	,	Comma	,
RBR	Adverb, comparative	<i>faster</i>	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	<i>fastest</i>	:	Mid-sentence punc	(: ; ... --)
RP	Particle	<i>up, off</i>			

Penn Treebank Tagset

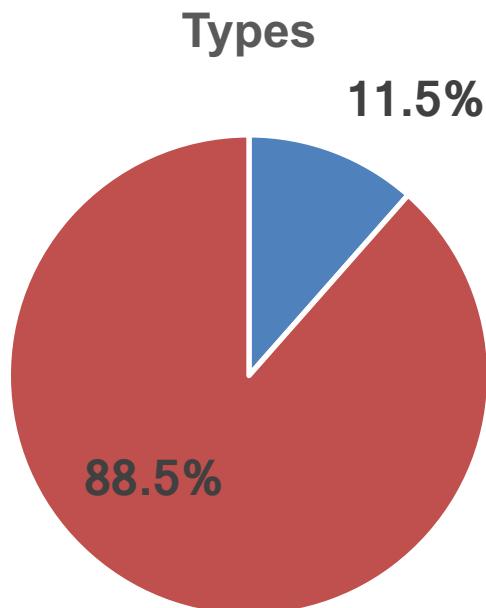
Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	<i>and, but, or</i>	SYM	Symbol	+%, &
CD	Cardinal number	<i>one, two, three</i>	TO	"to"	<i>to</i>
DT	Determiner	<i>a, the</i>	UH	Interjection	<i>ah, oops</i>
EX	Existential 'there'	<i>there</i>	VB	Verb, base form	<i>eat</i>
FW	Foreign word	<i>mea culpa</i>	VBD	Verb, past tense	<i>ate</i>
IN	Preposition/sub-conj	<i>of, in, by</i>	VBG	Verb, gerund	<i>eating</i>
JJ	Adjective	<i>yellow</i>	VBN	Verb, past participle	<i>eaten</i>
JJR	Adj., comparative	<i>bigger</i>	VBP	Verb, non-3sg pres	<i>eat</i>
JJS	Adj., superlative	<i>wildest</i>	VBZ	Verb, 3sg pres	<i>eats</i>
LS	List item marker	<i>1, 2, One</i>	WDT	Wh-determiner	<i>which, that</i>
MD	Modal	<i>can, should</i>	WP	Wh-pronoun	<i>what, who</i>
NN	Noun, sing. or mass	<i>llama</i>	WP\$	Possessive wh-	<i>whose</i>
NNS	Noun, plural	<i>llamas</i>	WRB	Wh-adverb	<i>how, where</i>
NNP	Proper noun, singular	<i>IBM</i>	\$	Dollar sign	\$
NNPS	Proper noun, plural	<i>Carolinas</i>	#	Pound sign	#
PDT	Predeterminer	<i>all, both</i>	"	Left quote	(‘ or “)
POS	Possessive ending	<i>'s</i>	"	Right quote	(‘ or ”)
PRP	Personal pronoun	<i>I, you, he</i>	(Left parenthesis	([, (, {, <)
PRP\$	Possessive pronoun	<i>your, one's</i>)	Right parenthesis	(],), }, >)
RB	Adverb	<i>quickly, never</i>	,	Comma	,
RBR	Adverb, comparative	<i>faster</i>	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	<i>fastest</i>	:	Mid-sentence punc	(: ; ... --)
RP	Particle	<i>up, off</i>			

Susan decided to run an errand while Joe went on his run.

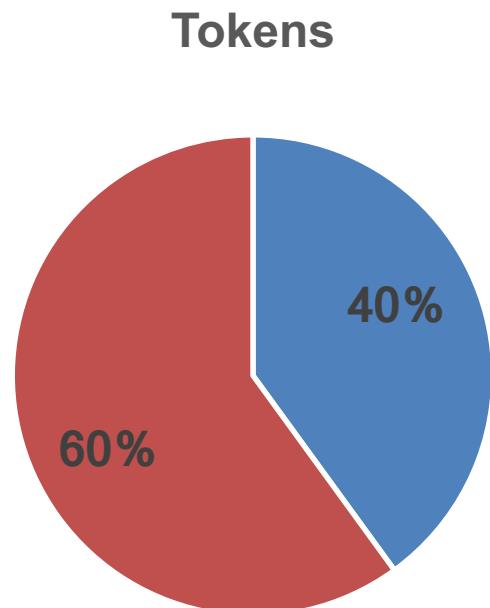
NNP VBD TO VB DT NN IN NNP VBD IN PRP\$ NN

How Ambiguous Is POS?

Based on the Brown corpus:



■ 1 POS ■ >1 POS



■ 1 POS ■ >1 POS

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.

Susan decided to jar some preserves.

NNP

How POS Taggers Work

Generally there is a sliding window.

Susan decided to jar some preserves.
NNP VBD

How POS Taggers Work

Generally there is a sliding window.

Susan decided to jar some preserves.
NNP VBD IN

How POS Taggers Work

Generally there is a sliding window.

Susan decided to jar some preserves.
NNP VBD TO VB

Notice the adjustment that just happened?

How POS Taggers Work

Generally there is a sliding window.

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

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.

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

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

And we can go backwards, too...

How POS Taggers Work

Backward-sliding window:

Susan decided to jar some **preserves.**
NNS

How POS Taggers Work

Backward-sliding window:

Susan decided to jar some preserves.
JJ NNS

How POS Taggers Work

Backward-sliding window:

Susan decided to jar some preserves.
NN JJ NNS

How POS Taggers Work

Backward-sliding window:

Susan decided **to jar some preserves.**
TO VB JJ NNS

Notice the adjustment that just happened?

How POS Taggers Work

Backward-sliding window:

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

How POS Taggers Work

Backward-sliding window:

Susan decided to jar some preserves.

NNP VBD TO VB JJ NNS

How POS Taggers Work

Backward-sliding window:

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

How POS Taggers Work

Backward-sliding window:

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

DataScience@SMU

Syntactic Analysis: POS Tagging (Continued)

Natural Language Processing

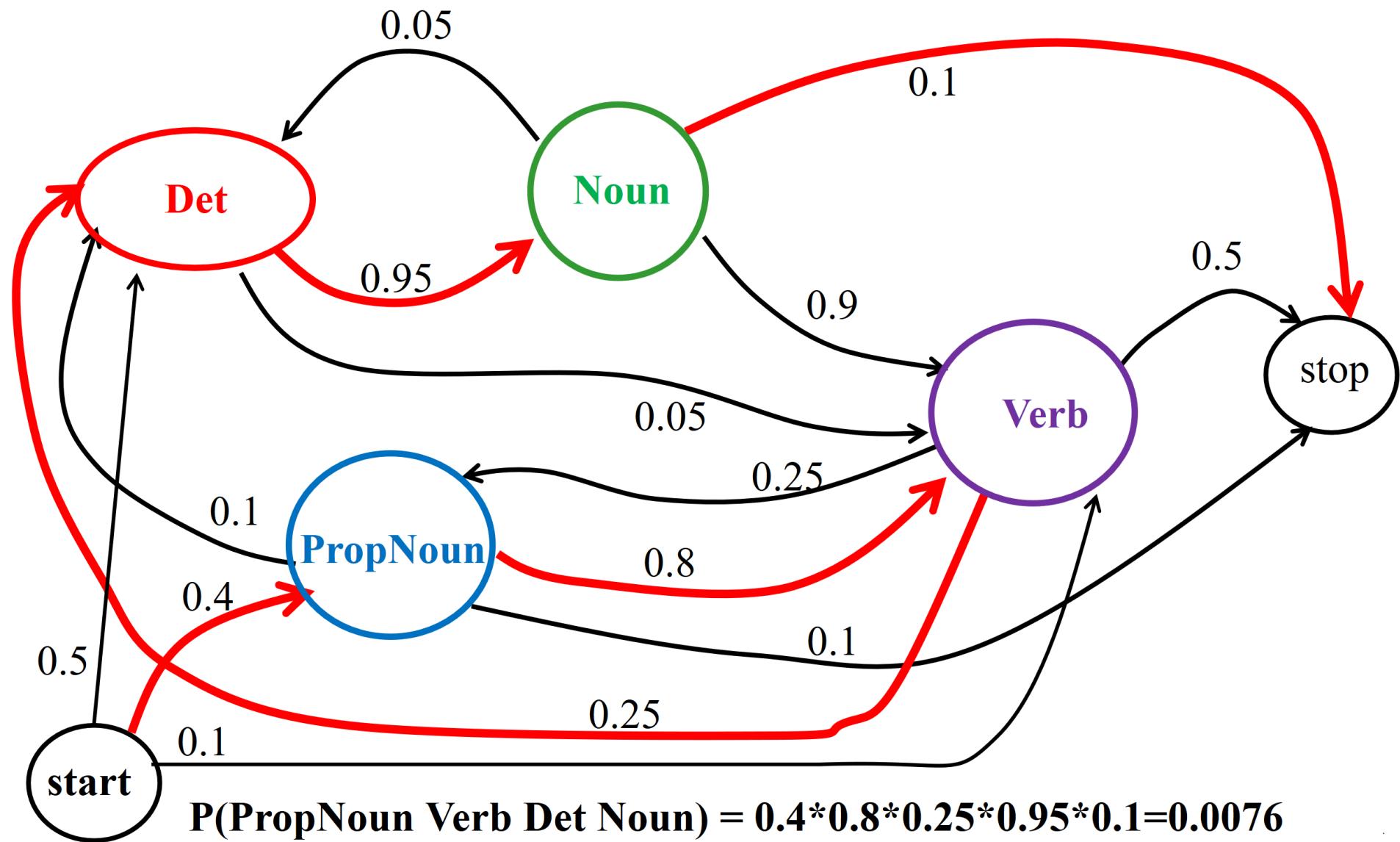
An Early Approach to Statistical POS Tagging

- 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%

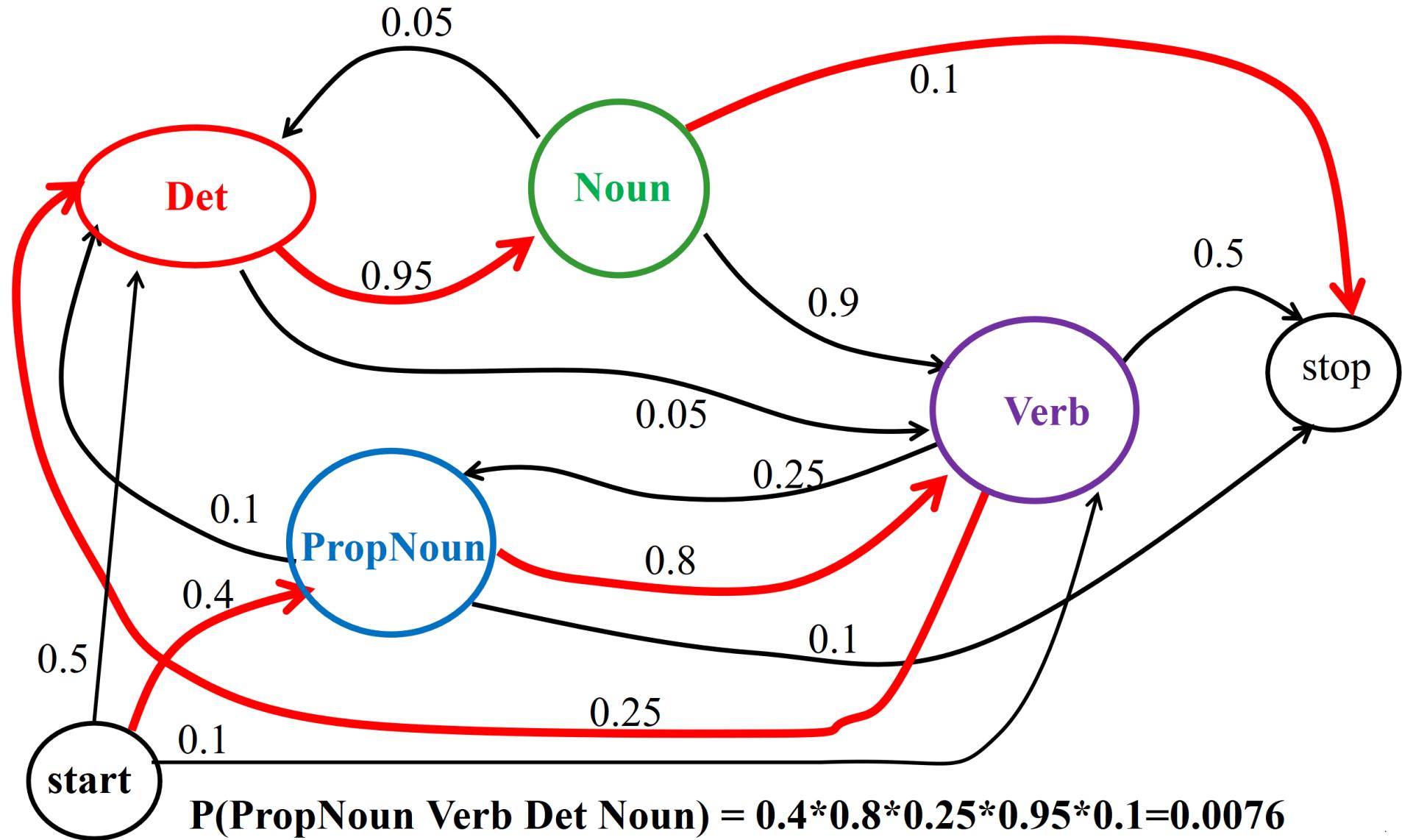
- After consulting all applicable tables for a token, select tag with the highest overall likelihood.

Example HMM for POS: “Mary saw a cat”



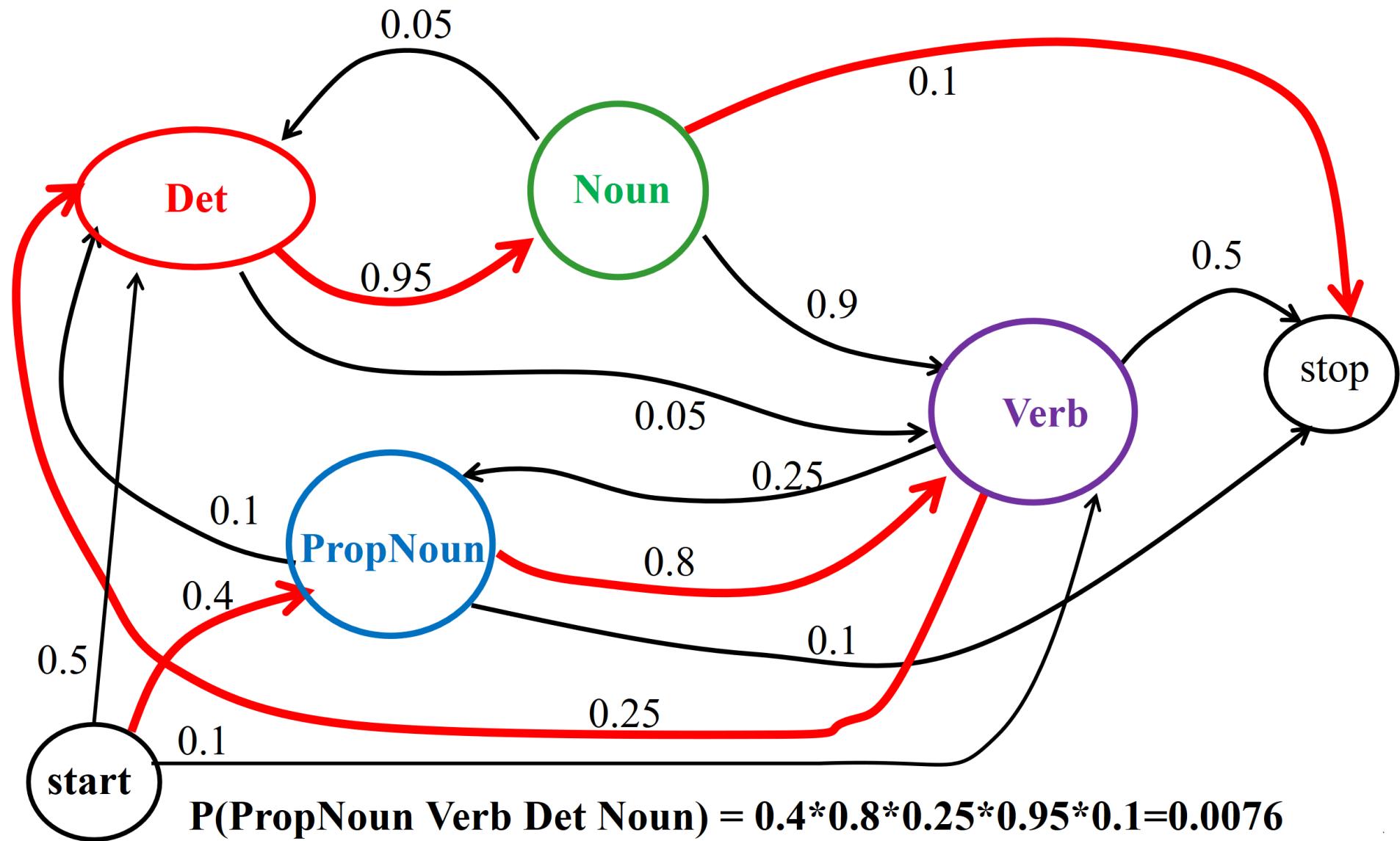
Consider that “saw” could be a noun.
Why does it not get labelled a noun here?

“Mary saw a cat”

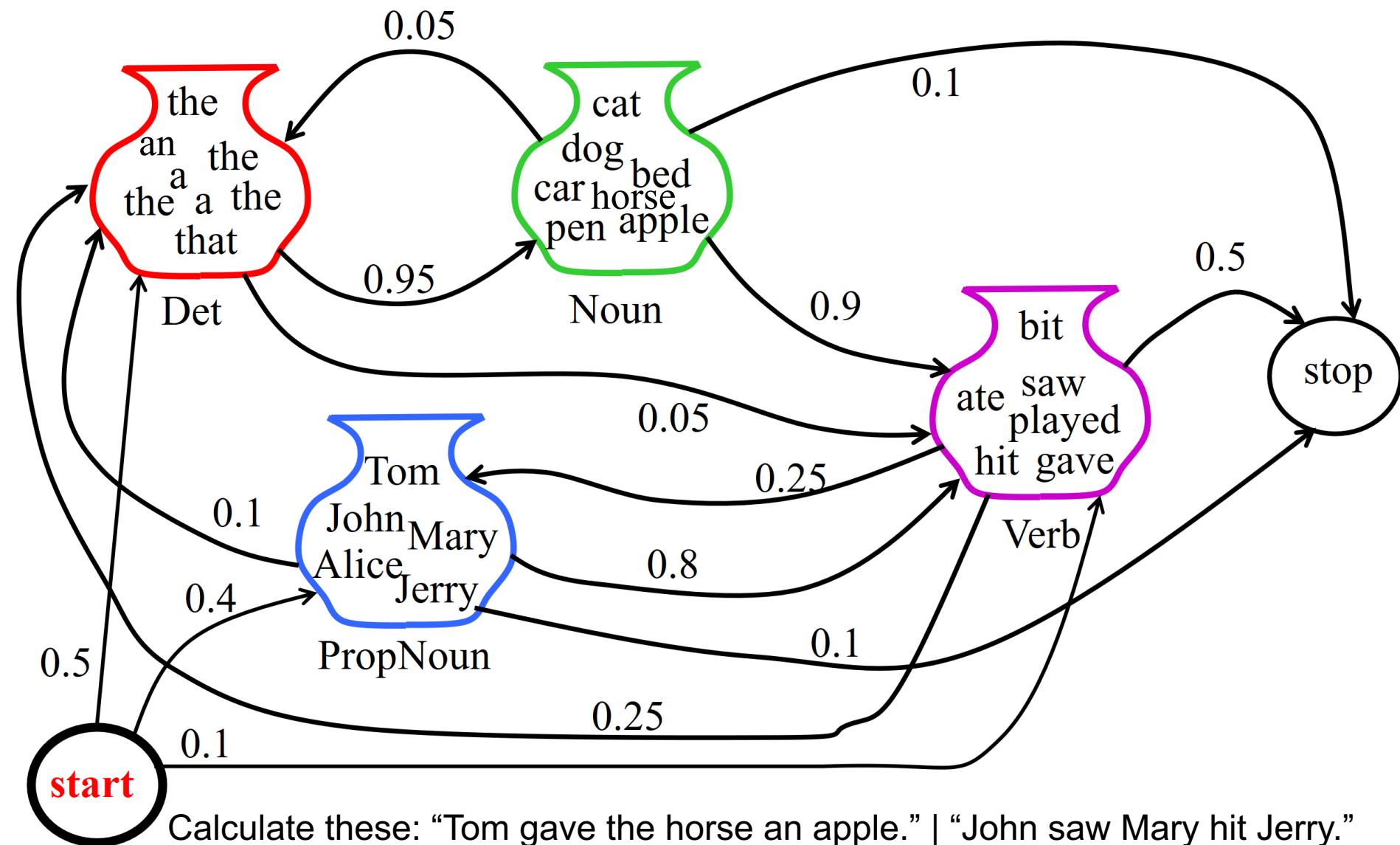


Consider that “cat” can actually be a verb.
Why does it not get labelled a verb here?

“Mary saw a cat”



Example HMM for POS



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

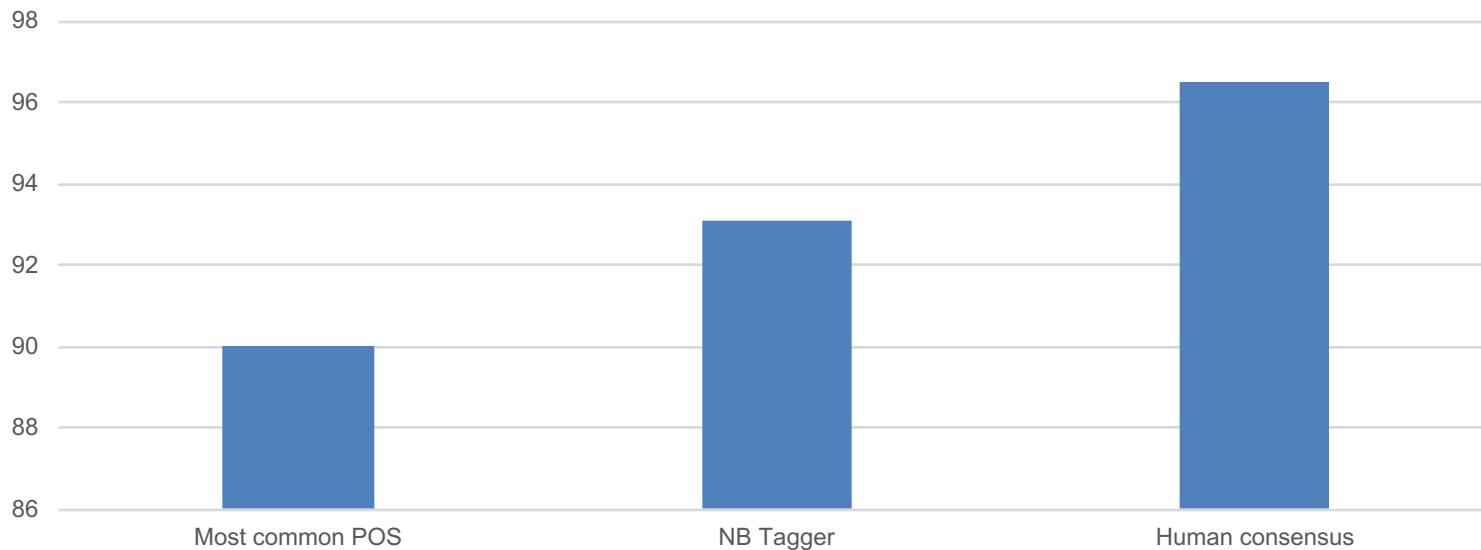


How Good Is It?

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.

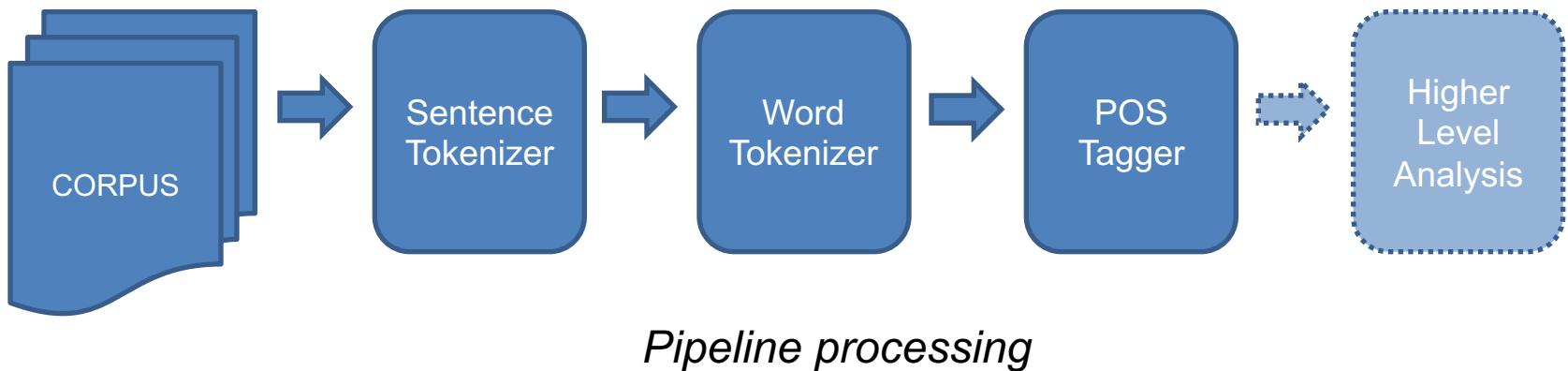
Comparative Accuracy



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.



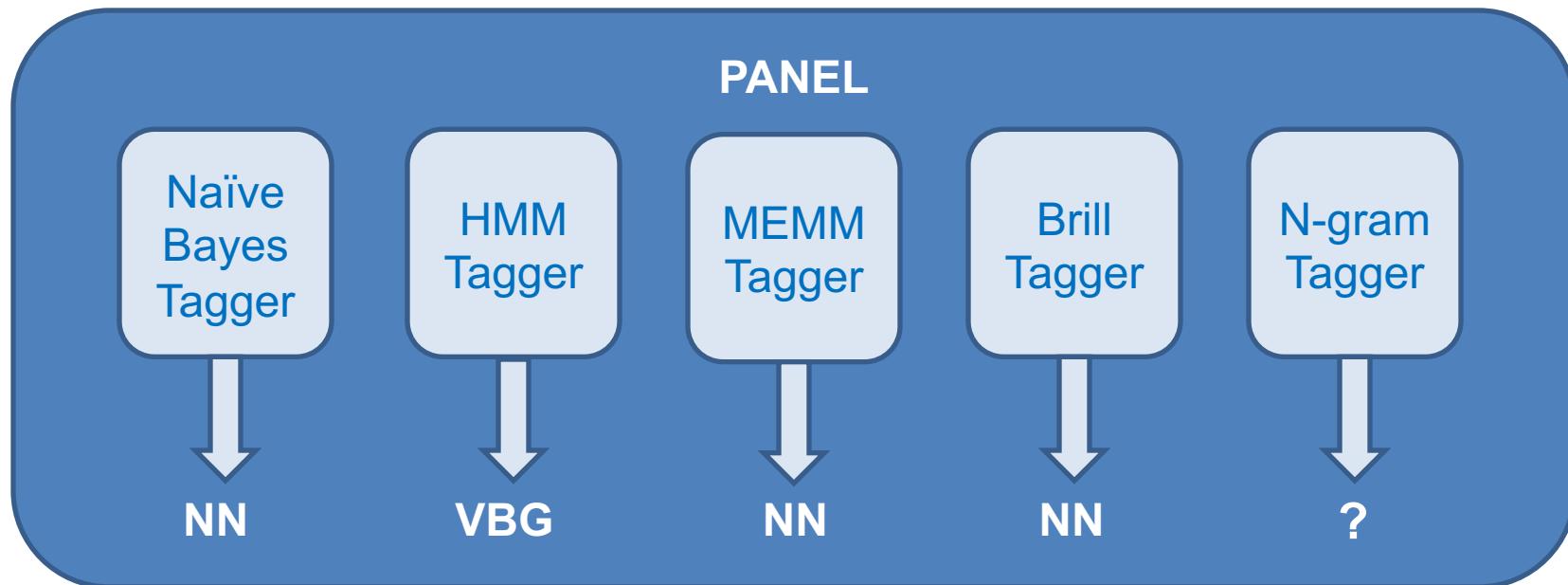
Combining Taggers

Bonus question:

How would you systematically combine three or more POS taggers?

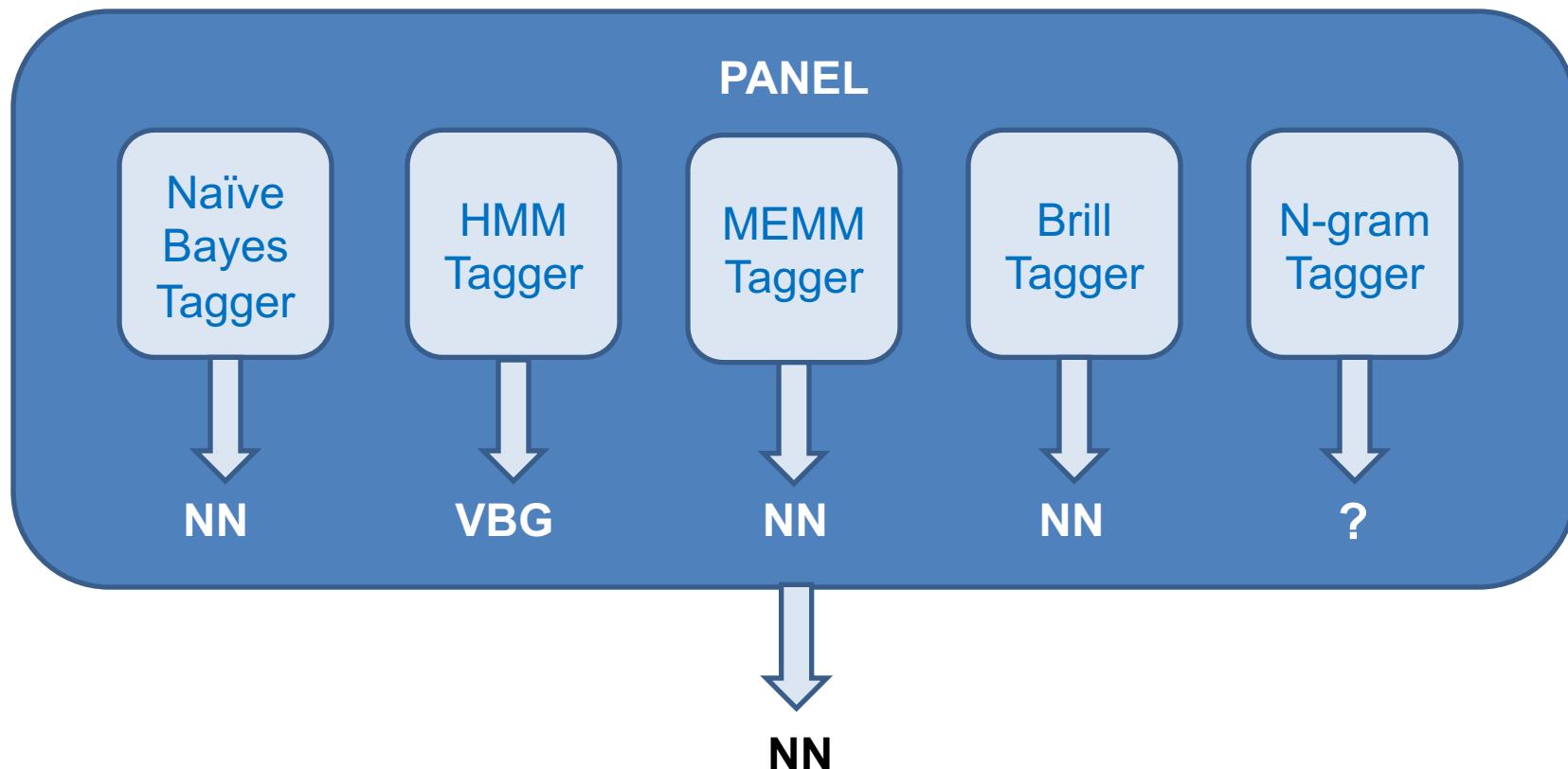
Combining Taggers

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



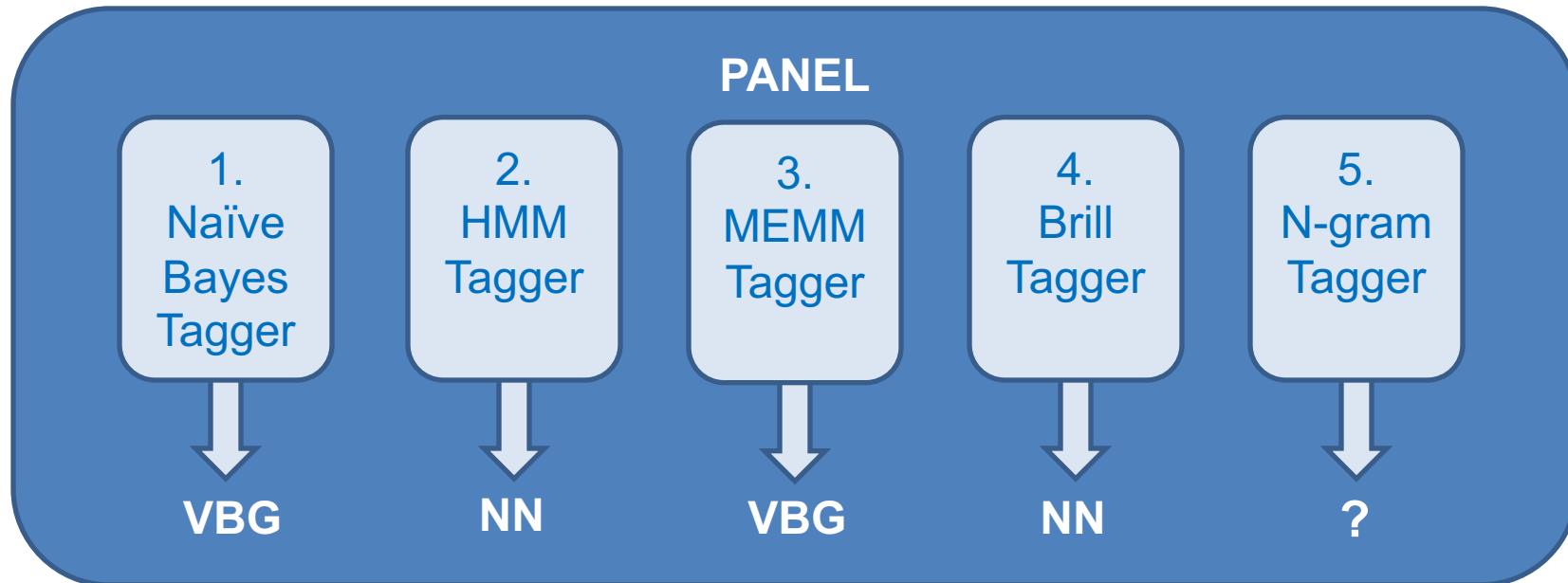
Combining Taggers

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



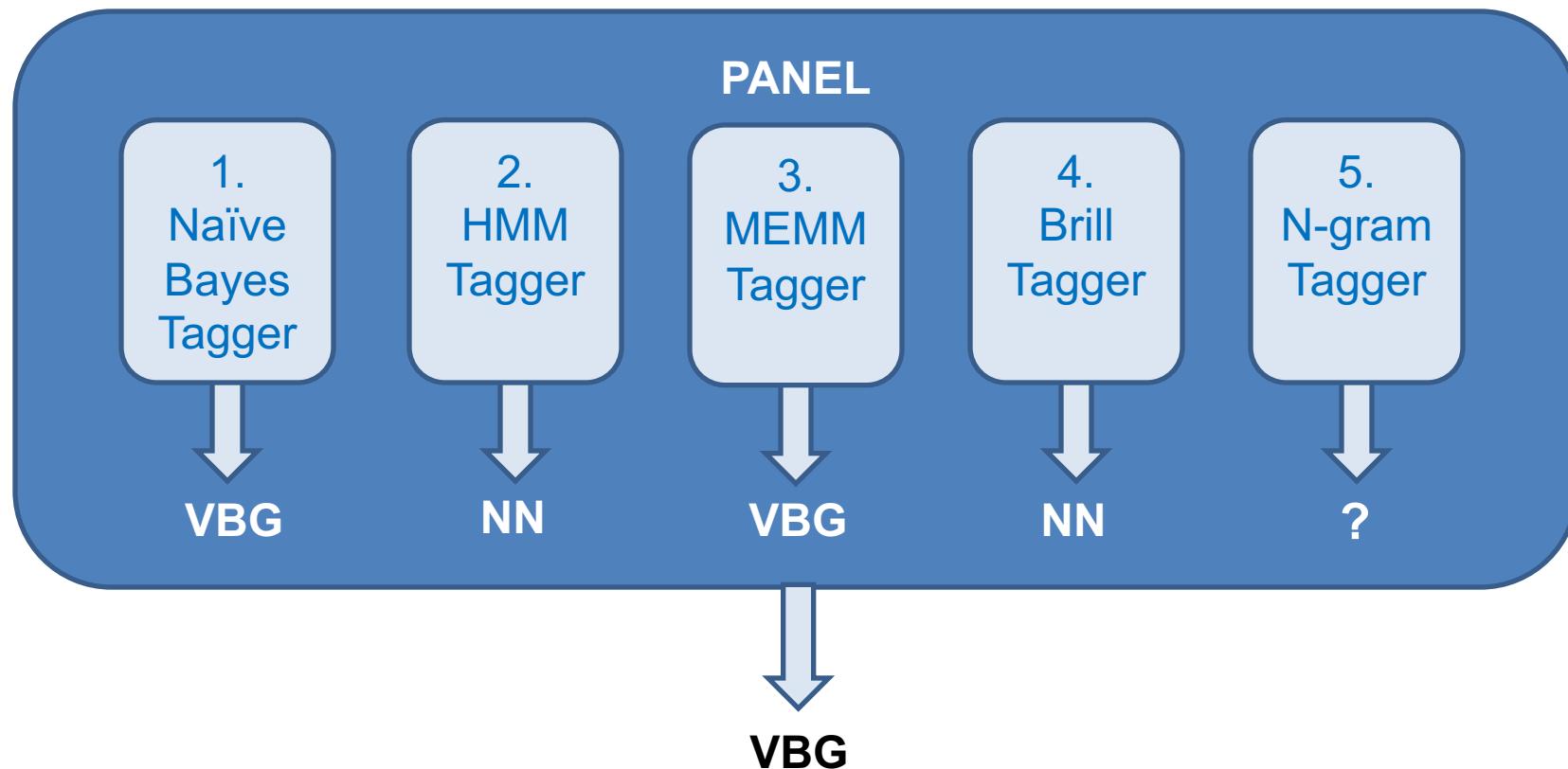
Combining Taggers

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

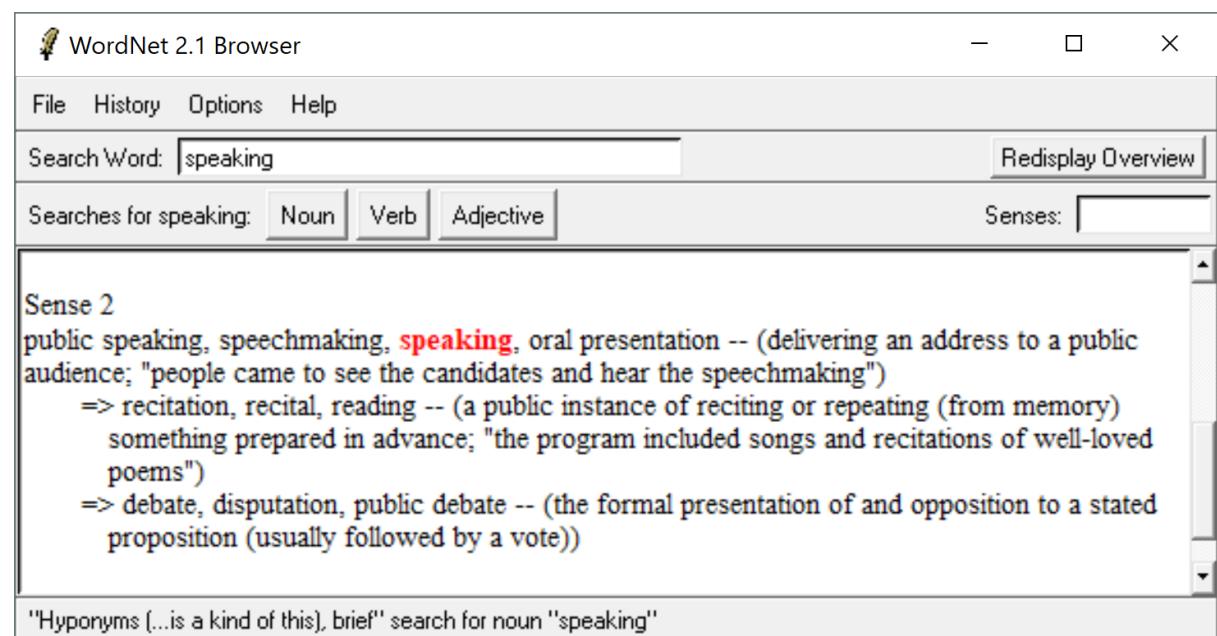
Knowing the likely POS for a keyword in a search query helps us when considering synonyms/hyponyms to substitute as related search terms.

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

Query: *lessons on speaking*

See also:

lessons on public debate



Using POS Tags with WordNet

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

The screenshot shows the WordNet 2.1 Browser interface. The title bar reads "WordNet 2.1 Browser". The menu bar includes "File", "History", "Options", and "Help". The toolbar has a "Search Word:" field containing "speaking", a "Redisplay Overview" button, and buttons for "Noun", "Verb", and "Adjective". Below the toolbar, it says "Searches for speaking: Noun Verb Adjective". The main window displays "Sense 1" for the verb "speaking". The sense definition is: "talk, **speak**, utter, mouth, verbalize, verbalise -- (express in speech; "She talks a lot of nonsense"; "This depressed patient does not verbalize")". A large list of troponyms follows, each preceded by a right-pointing arrow (=>):

- => read -- (look at, interpret, and say out loud something that is written or printed; "The King will read the proclamation at noon")
- => vocalize, vocalise, phonate -- (utter speech sounds)
- => troll -- (speak or recite rapidly or in a rolling voice)
- => begin -- (begin to speak or say; "Now listen, friends," he began)
- => lip off, shoot one's mouth off -- (speak spontaneously and without restraint; "She always shoots her mouth off and says things she later regrets")
- => shout -- (utter in a loud voice; talk in a loud voice (usually denoting characteristic manner of speaking); "My grandmother is hard of hearing--you'll have to shout")
- => whisper -- (speak softly; in a low voice)
- => peep -- (speak in a hesitant and high-pitched tone of voice)
- => speak up -- (speak louder; raise one's voice; "The audience asked the lecturer to please speak up")
- => snap, snarl -- (utter in an angry, sharp, or abrupt tone; "The sales clerk snapped a reply at the angry customer"; "The guard snarled at us")
- => enthuse -- (utter with enthusiasm)
- => speak in tongues -- (speak unintelligibly in or as if in religious ecstasy; "The parishioners spoke in tongues")
- => swallow -- (utter indistinctly; "She swallowed the last words of her speech")
- => verbalize, verbalise -- (be verbose; "This lawyer verbalizes and is rather tedious")
- => whiff -- (utter with a puff of air; "whiff out a prayer")
- => talk of, talk about -- (discuss or mention; "They spoke of many things")
- => blubber, blubber out -- (utter while crying)
- => drone, drone on -- (talk in a monotonous voice)
- => bumble, stutter, stammer, falter -- (speak haltingly; "The speaker faltered when he saw his opponent enter the room")
- => rasp -- (utter in a grating voice)
- => blurt out, blurt, blunder out, blunder, ejaculate -- (utter impulsively; "He blurted out the secret"; "He blundered his stupid ideas")

"Troponyms (particular ways to...), brief" search for verb "speaking"

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

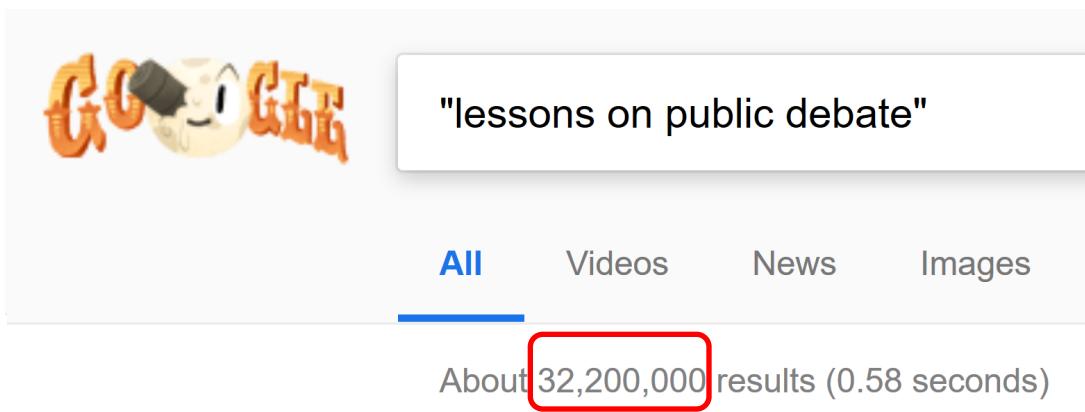
Using POS Tags with WordNet

Here's what saves us:



A screenshot of a Google search results page. The search bar contains the query "lessons on whispering". Below the search bar, there are four tabs: All (which is underlined in blue), Videos, Images, and Shopping. Underneath the tabs, the text "About 3 results (0.29 seconds)" is displayed. A red box highlights the number "3".

VS.



A screenshot of a Google search results page. The search bar contains the query "lessons on public debate". Below the search bar, there are four tabs: All (underlined in blue), Videos, News, and Images. Underneath the tabs, the text "About 32,200,000 results (0.58 seconds)" is displayed. A red box highlights the number "32,200,000".

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**



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

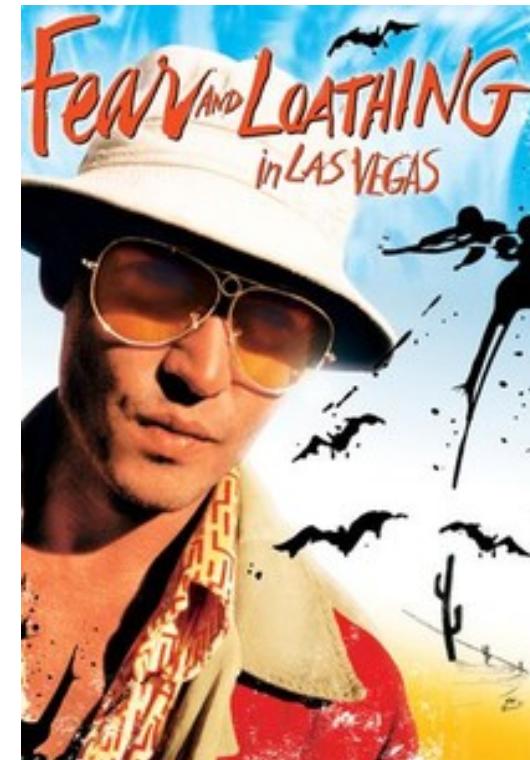
Here's how it regenerates if we eliminate verbs:



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.”

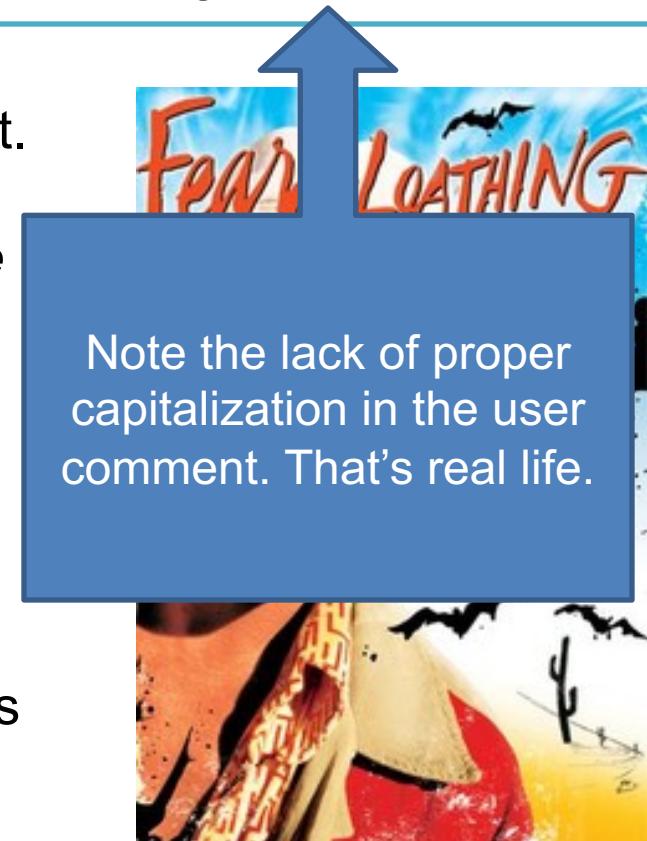
- Clearly (to us humans) this user comment has overall *positive* sentiment.
- But because of the negative words, it would likely register as overall *negative sentiment* to a basic, weighted-vocabulary sentiment analyzer.
- That’s because a *stemmer* would be invoked to treat “loathe, loathed, loathing” 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