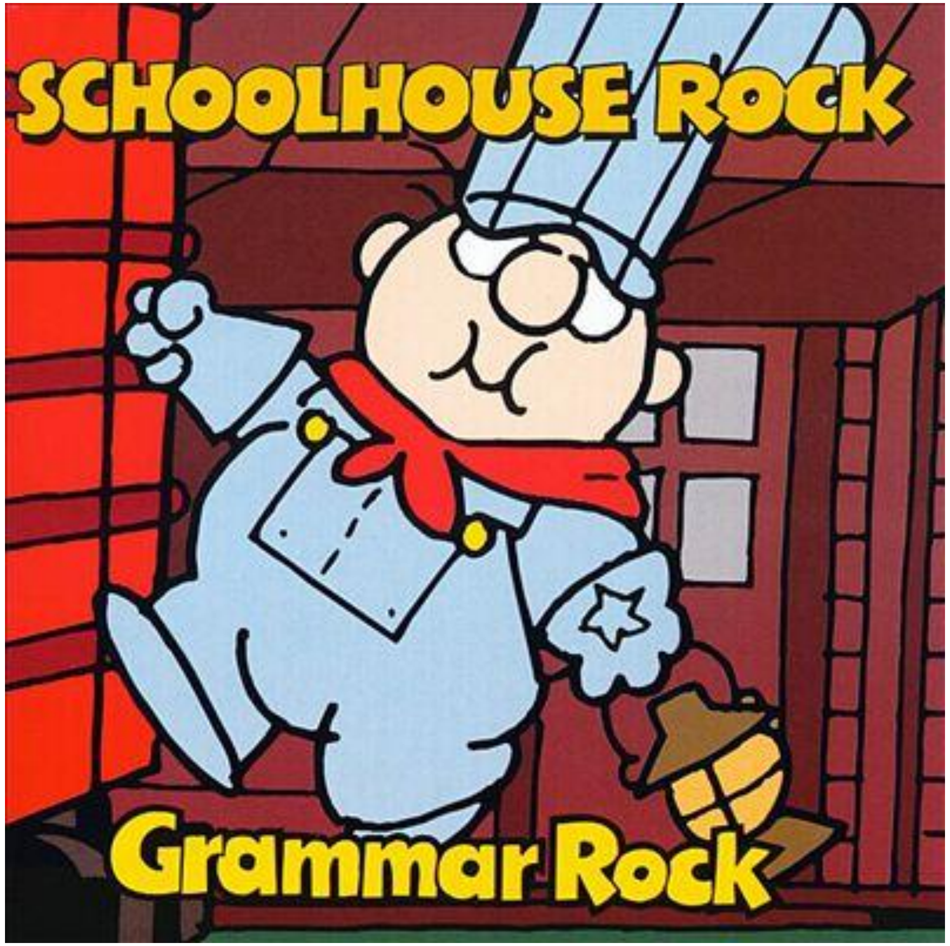
Lecture 6

POS Tagging

Hit record

Basic Parts of speech (POS) from school

- Noun (e.g. The *dog* barked.)
- Verb (e.g. Tim *jumped* up.)
- Pronoun (e.g. *He* sat down.)
- Preposition (e.g. He hid *under* the covers.)
- Adverb (e.g. She *slowly* stood up.)
- Conjunction (e.g. He talked *and* ate.)
- Participle (e.g. She saw John *eating)*
- Article (e.g. *The* cat ate *a* mouse)



Extended over time

- Penn Treebank (Marcus, et al., 1993) : 45
- Brown corpus (Francis, 1979) : 87
- C7 (Garside, et al., 1997) : 146

Significance of POS (aka: word classes, tag sets, lexical tags)

the amount of information they give about a word and its neighbors

POS and WSD


S: (n) rear, back (the side that goes last or is not normally seen)

- (n) He wrote the date on the *back* of the photograph.

- (v) *Back* the car into the driveway.

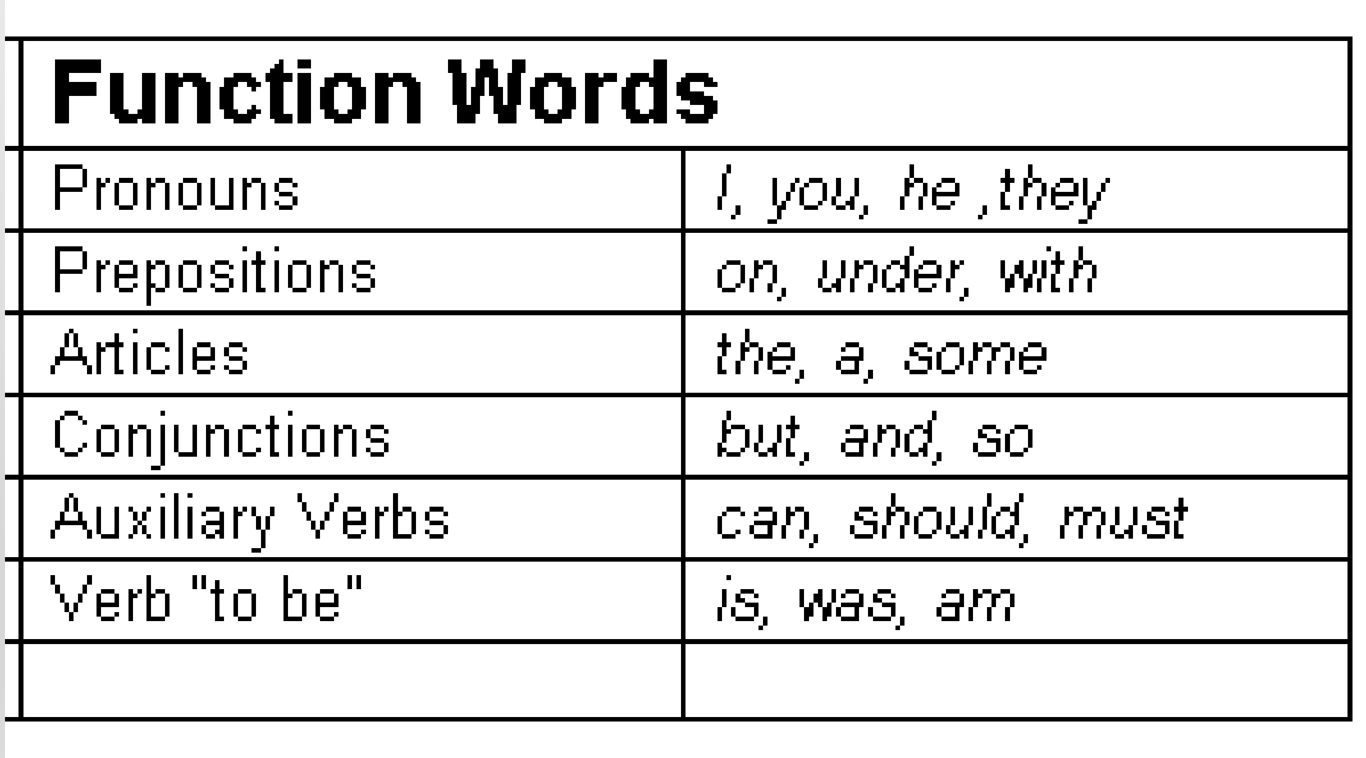

S: (v) back (travel backward)

2 broad categories

- Closed class
- Open class

Close Class

- Fixed membership (e.g. prepositions, articles)

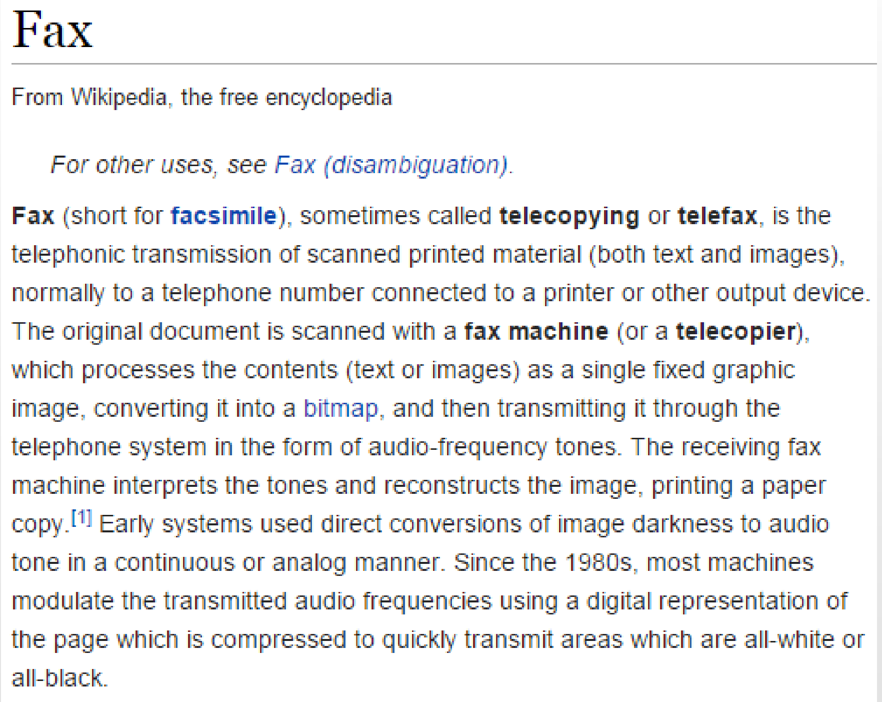


Function Words	
Pronouns	*I, you, he ,they*
Prepositions	*on, under, with*
Articles	*the, a, some*
Conjunctions	*but, and, so*
Auxiliary Verbs	*can, should, must*
Verb "to be"	*is, was, am*

We don’t introduce new function words into the language.

Open Class

- always adding (e.g. nouns, verbs)



Fax

From Wikipedia, the free encyclopedia

For other uses, see Fax (disambiguation).

Fax (short for **facsimile**), sometimes called **telecopying** or **telefax**, is the telephonic transmission of scanned printed material (both text and images), normally to a telephone number connected to a printer or other output device. The original document is scanned with a **fax machine** (or a **telecopier**), which processes the contents (text or images) as a single fixed graphic image, converting it into a bitmap, and then transmitting it through the telephone system in the form of audio-frequency tones. The receiving fax machine interprets the tones and reconstructs the image, printing a paper copy.[1] Early systems used direct conversions of image darkness to audio tone in a continuous or analog manner. Since the 1980s, most machines modulate the transmitted audio frequencies using a digital representation of the page which is compressed to quickly transmit areas which are all-white or all-black.

n. fax: a duplicator that transmits the copy by wire or radio

Over time

v. fax: send something via a fax machine

Nouns: person, place or thing

Noun

Proper Noun

Common Noun

Mass Noun
homogenous group

we don't talk about *three snows*

Count Noun
allow grammatical enumeration

one goat, two goats, three goats, four

Verbs: actions or processes

To draw

To eat

To debate

Morphological Forms

She eats	She ate	She **will** eat
She is eating	She **was** eating when my father came	She **will be** eating when my father comes
She **has** eaten	She **had** eaten before you came	She **will have** eaten before you come
She **has been** eating	She **had been** eating	She **will have been** eating

Adjectives: describe properties or qualities

Color: white and black

Age: old and young

Value: good and bad

Adverbs: bit of a hodgepodge

Adverbs

Directional

- Home
- Here
- Downhill

Degree

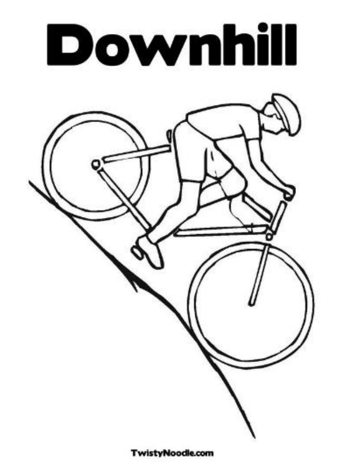
- Extremely
- Very
- Somewhat

Manner

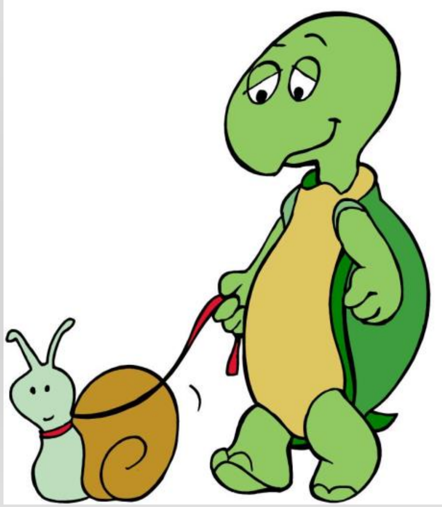
- Slowly
- Delicately
- Silkily

Temporal

- Yesterday
- Monday
- Today




Somewhat_



POS Tagging

- POS tagging
    - is the process of assigning a part of speech to each word in a corpus

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

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

Choosing a Tag set

- To do POS tagging, first need to choose a set of tags
- Could pick very coarse (small) tagsets
  - N, V, Adj, Adv.
- More commonly used: Brown Corpus (Francis & Kucera ‘82), 1M words, 87 tags – more informative but more difficult to tag
- Most commonly used: Penn Treebank: hand-annotated corpus of *Wall Street Journal*, 1M words, 45-46 subset

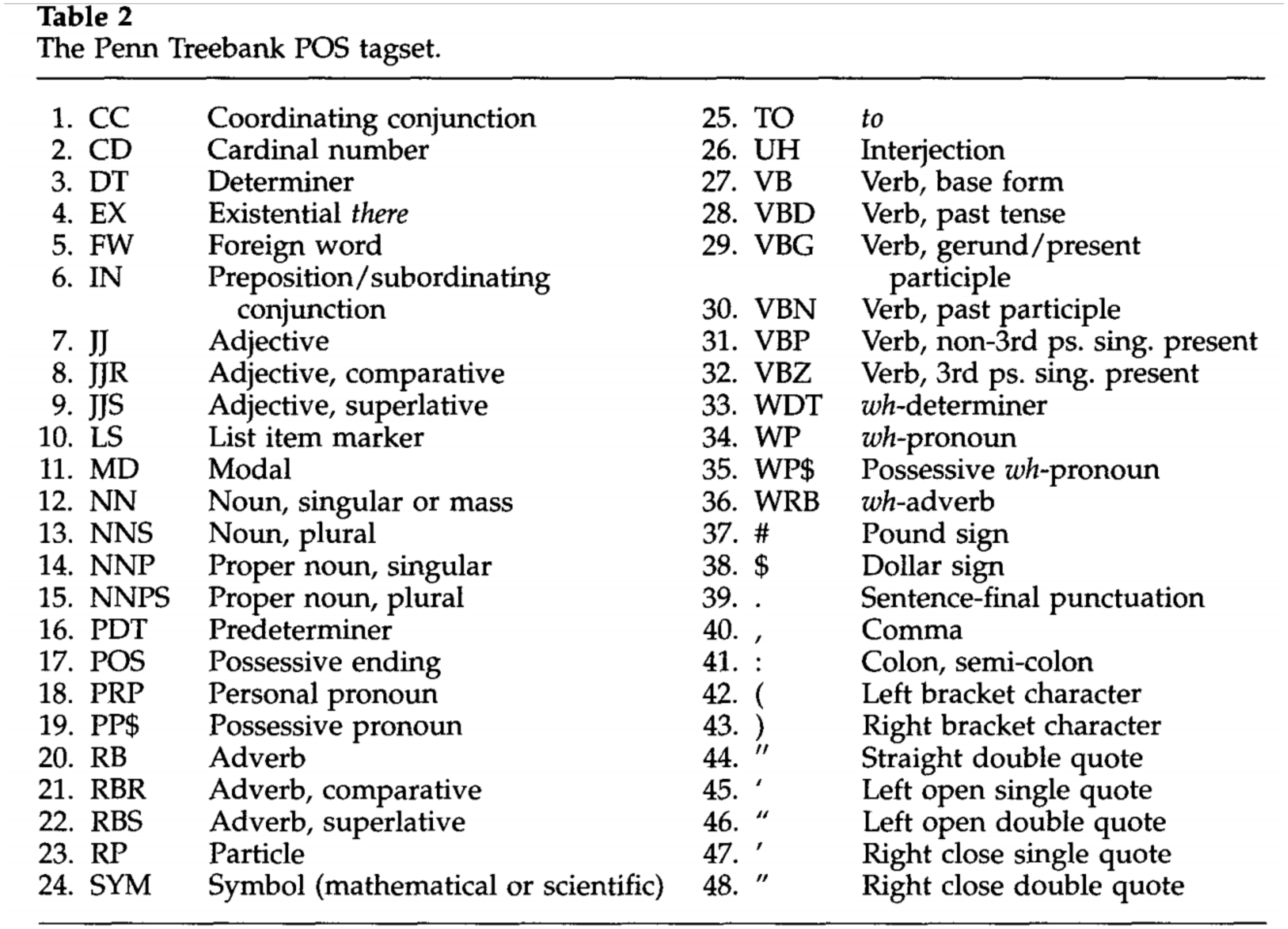


Table 2
The Penn Treebank POS tagset.

1.	CC	Coordinating conjunction	25.	TO	*to*
2.	CD	Cardinal number	26.	UH	Interjection
3.	DT	Determiner	27.	VB	Verb, base form
4.	EX	Existential *there*	28.	VBD	Verb, past tense
5.	FW	Foreign word	29.	VBG	Verb, gerund/present participle
6.	IN	Preposition/subordinating conjunction	30.	VBN	Verb, past participle
7.	JJ	Adjective	31.	VBP	Verb, non-3rd ps. sing. present
8.	JJR	Adjective, comparative	32.	VBZ	Verb, 3rd ps. sing. present
9.	JJS	Adjective, superlative	33.	WDT	*wh*-determiner
10.	LS	List item marker	34.	WP	*wh*-pronoun
11.	MD	Modal	35.	WP$	Possessive *wh*-pronoun
12.	NN	Noun, singular or mass	36.	WRB	*wh*-adverb
13.	NNS	Noun, plural	37.	#	Pound sign
14.	NNP	Proper noun, singular	38.	$	Dollar sign
15.	NNPS	Proper noun, plural	39.	.	Sentence-final punctuation
16.	PDT	Predeterminer	40.	,	Comma
17.	POS	Possessive ending	41.	:	Colon, semi-colon
18.	PRP	Personal pronoun	42.	(	Left bracket character
19.	PP$	Possessive pronoun	43.	)	Right bracket character
20.	RB	Adverb	44.	″	Straight double quote
21.	RBR	Adverb, comparative	45.	‘	Left open single quote
22.	RBS	Adverb, superlative	46.	“	Left open double quote
23.	RP	Particle	47.	’	Right close single quote
24.	SYM	Symbol (mathematical or scientific)	48.	″	Right close double quote

Using the Penn Tree Bank

- The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.
- Prepositions and subordinating conjunctions marked IN (“although/IN I/PRP..”)
- Except the preposition/complementizer “to” is just marked “TO”
- ***NB: PRP$ (possessive pronoun) vs. $***

Tagging can be hard for humans and machines

- *Around* can refer to an

  RB: adverb IN: preposition RP: particle

- Mrs./NNP Shaefer/NNP never/RB got/VBD **around/RP** to/TO joining/VBG

- All/Dt we/PRP gotta/VBN do/VB is/VBZ go/VB **around/IN** the/DT corner/NN

- Chateau/NNP Petrus/NNP costs/VBZ **around/RB** 250/CD

Tag Ambiguity

- Words often have more than one POS: *back*
  - The *back* door = JJ
  - On my *back* = NN
  - Win the voters *back* = RB
  - Promised to *back* the bill = VB
- The POS tagging problem is ***to determine the POS tag for a particular instance of a word***

Tagging whole sentences is hard

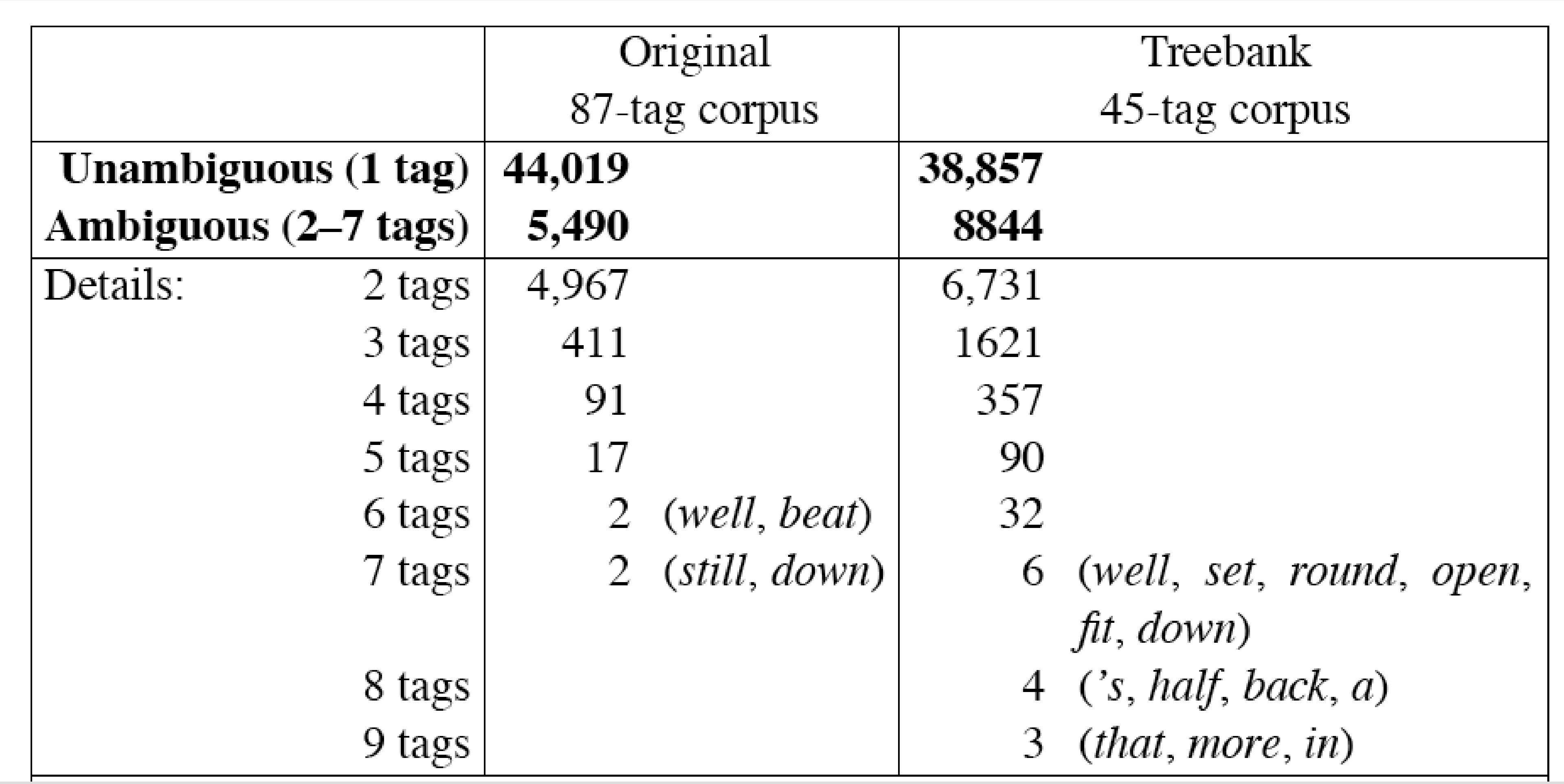
- Ambiguous POS contexts

Time flies like an arrow.

- Possible POS assignments
  - Time/[V,N] flies/[V,N] like/[V,Prep] an/Det arrow/N
  - Time/N flies/V like/Prep an/Det arrow/N
  - Time/V flies/N like/Prep an/Det arrow/N
  - Time/N flies/N like/V an/Det arrow/N

  …..

Scale of the Ambiguity Problem



		Original 87-tag corpus	Treebank 45-tag corpus
Unambiguous (1 tag)		**44,019**	**38,857**
Ambiguous (2–7 tags)		**5,490**	**8844**
Details:	2 tags	4,967	6,731
	3 tags	411	1621
	4 tags	91	357
	5 tags	17	90
	6 tags	2 (*well*, *beat*)	32
	7 tags	2 (*still*, *down*)	6 (*well*, *set*, *round*, *open*, *fit*, *down*)
	8 tags		4 (*'s*, *half*, *back*, *a*)
	9 tags		3 (*that*, *more*, *in*)

How do we disambiguate?

- Many words have only one POS tag (e.g. is, Mary, very, smallest)
- Others have a single ***most likely*** tag (e.g. a, dog)
- Tags also tend to *co-occur* regularly with other tags (e.g. Det, N)
- In addition to conditional probabilities of words $P(w_1|w_{n-1})$, we can look at POS likelihoods ($P(t_1|t_{n-1})$) to disambiguate sentences and to assess sentence likelihoods

Two classes of tagging algorithms

- Rule based taggers
  - Use a large database of hand written rules that specify the POS of words
    - And if a word is ambiguous what POS typically follows the previous POS

- Stochastic taggers
  - Use a training corpus to compute the probability of a given word having a given tag in a given context

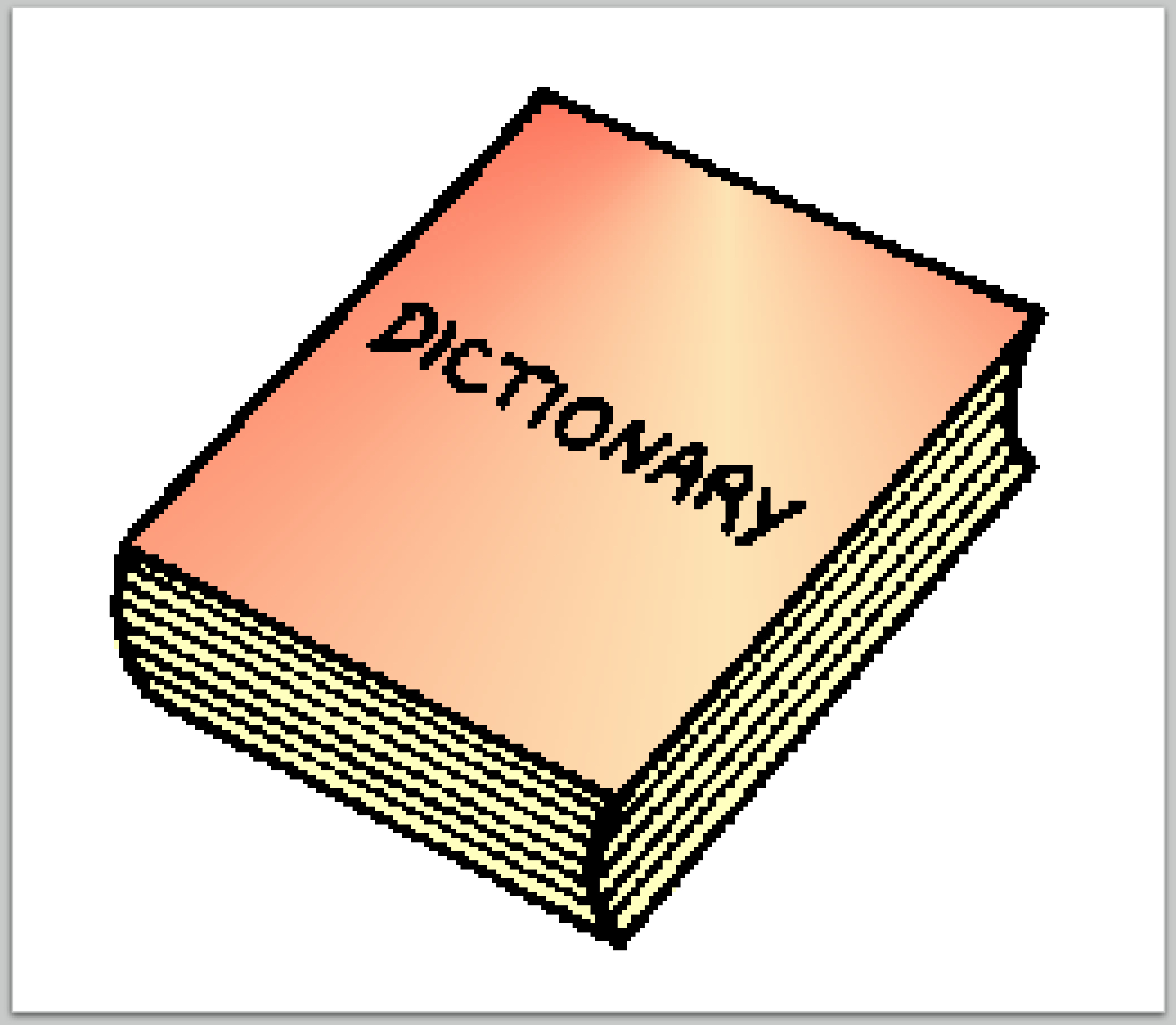
Rule-based POS tagging

- Earliest Algorithms had a 2 stage approach:
    - Stage 1: dictionary to assign each word a list of potential POS tags
    - Stage 2: list of rules to winnow down the list to a single POS

(Harris, 1962; Klein and Simmons, 1963; Greene and Rubin, 1971)

POS Dictionary

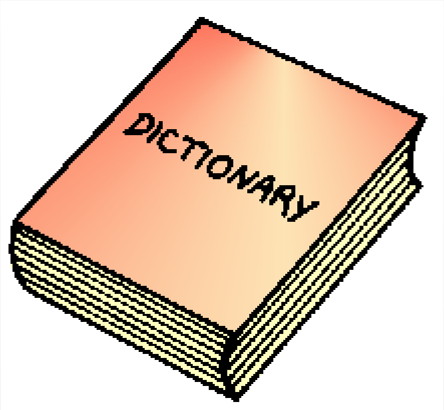
- she: PRP
- promised: VBN,VBD
- to: TO
- back: VB, JJ, RB, NN
- the: DT
- bill: NN, VB
- …
- for the ~100,000 words of English



POS Rules

E.g., *Eliminate VBN if VBD is an option when VBN|VBD follows "<start> PRP"*

She promised
to back the bill.

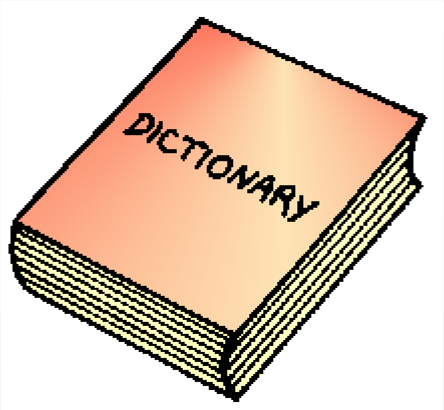


			NN		
			RB		
	VBN		JJ		VB
PRP	VBD	TO	VB	DT	NN
She	**promised**	**to**	**back**	**the**	**bill**

E.g., *Eliminate VBN if VBD is an option when VBN|VBD follows "<start> PRP"*

			NN		
			RB		
			JJ		VB
PRP	VBD	TO	VB	DT	NN
She	**promised**	**to**	**back**	**the**	**bill**

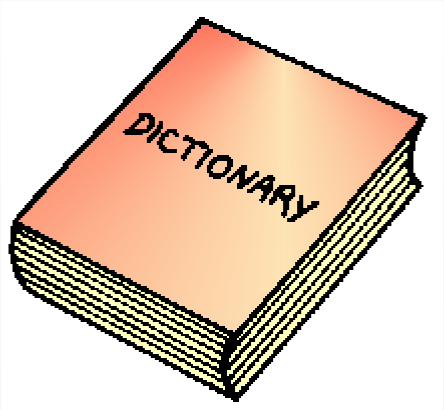
She promised
to back the bill.



			NN		
			RB		
	VBN		JJ		VB
PRP	VBD	TO	VB	DT	NN
She	**promised**	**to**	**back**	**the**	**bill**

			NN		
			RB		
			JJ		VB
PRP	VBD	TO	VB	DT	NN
She	**promised**	**to**	**back**	**the**	**bill**

She promised
to back the bill.



			NN		
			RB		
	VBN		JJ		VB
PRP	VBD	TO	VB	DT	NN
She	**promised**	**to**	**back**	**the**	**bill**

PRP	VBD	TO	VB	DT	NN
She	**promised**	**to**	**back**	**the**	**bill**

Modern rule sets

- Similar but have larger set of rules

- Example taggers:
  - EngCG tagger (Voutilainen, 1995, 1999)
  - Constraing Grammer (Karlsson et al., 1995)

Stochastic POS Tagger

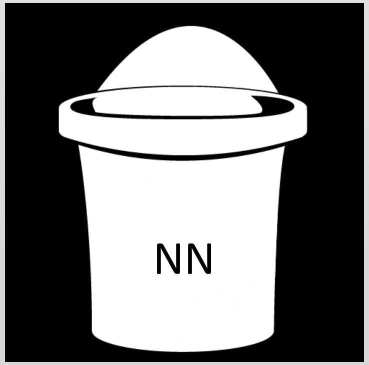
- Use probability information found in a training corpus

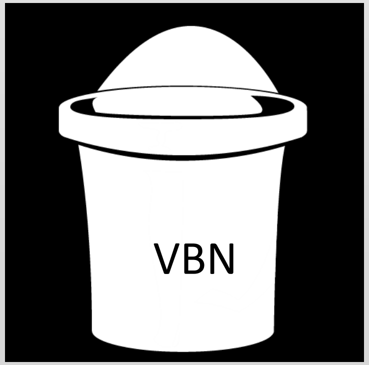
- Use of probabilities in POS Tagging is old:
    - Stolz, et al. 1965
    - Bahl and Mercer, 1976
    - Marshal, 1983
    - Garside, 1987
    - Church, 1988
    - DeRose, 1988

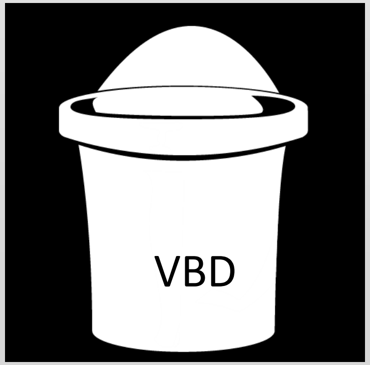
Treat POS tagging as a Classification Task

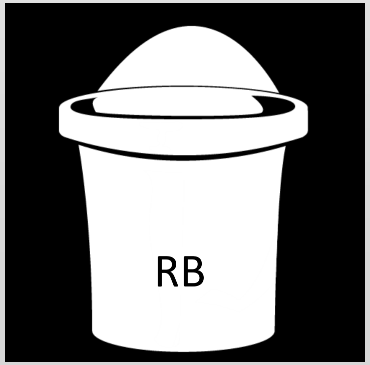
She	promised	to	back	the	bill

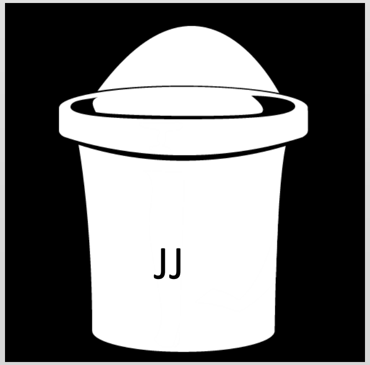
Classification: the goal is to determine which bucket each word is from







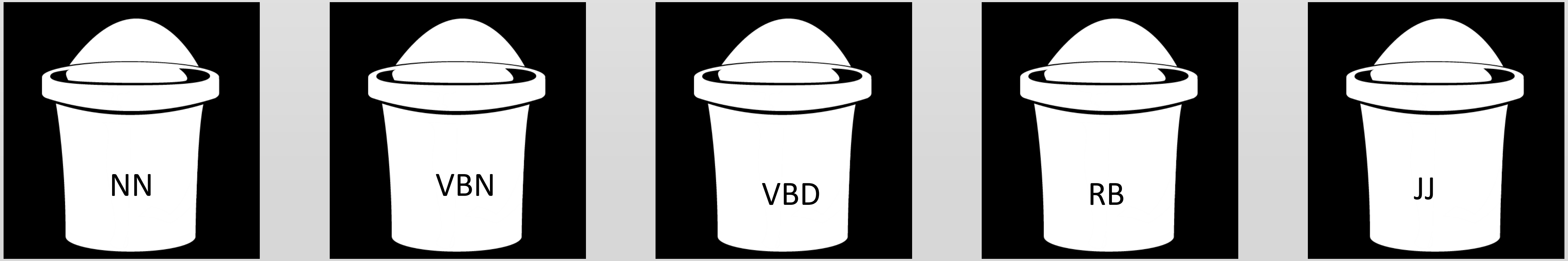




Modification: Sequence Classification Task

She	promised	to	back	the	bill

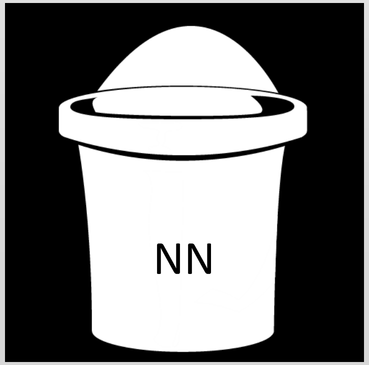
Sequence classification: the best sequence of tags that corresponds to this sequence of words

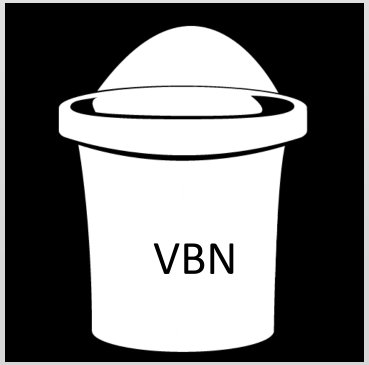


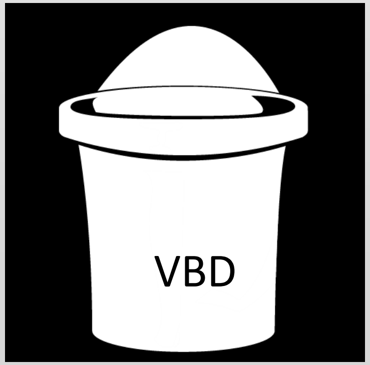
Modification: Sequence Classification Task

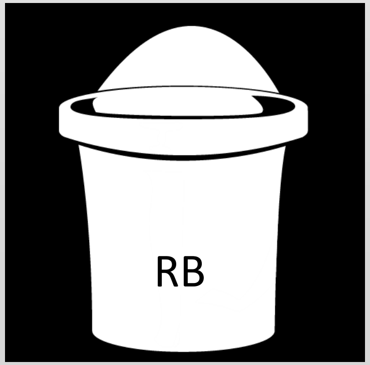
She	promised	to	back	the	bill

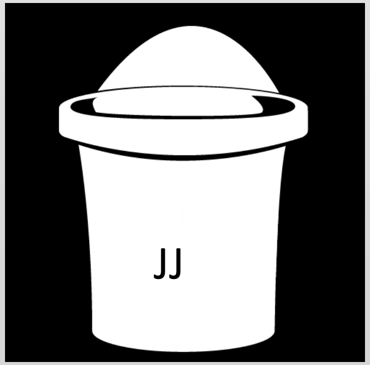
So this means that the probability of tag relies on the previous tags











HMM

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

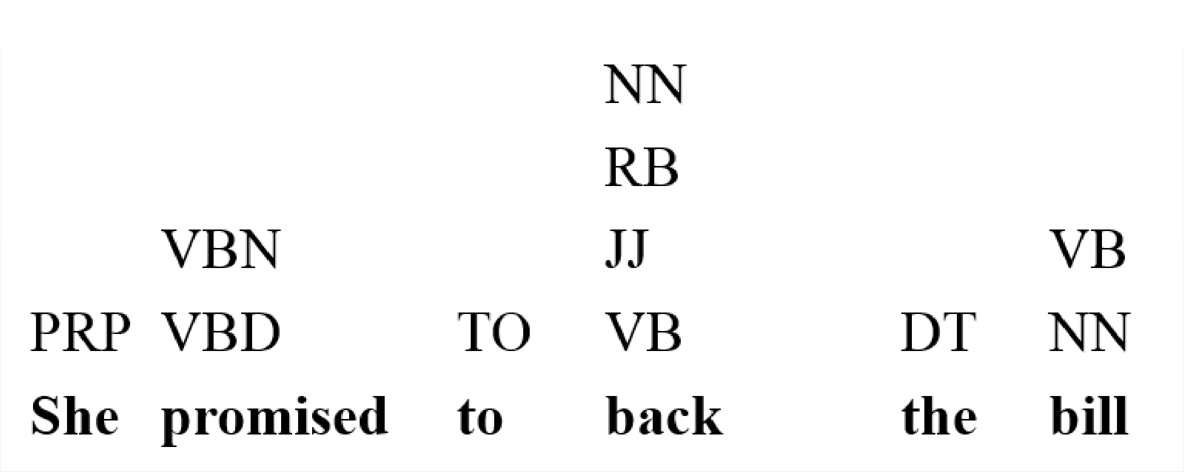
w_1^n = sequence of *n* words

t_1^n = sequence of *n* tags

$\hat{t}_1^n$ =estimate of the correct tag sequence

Example

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



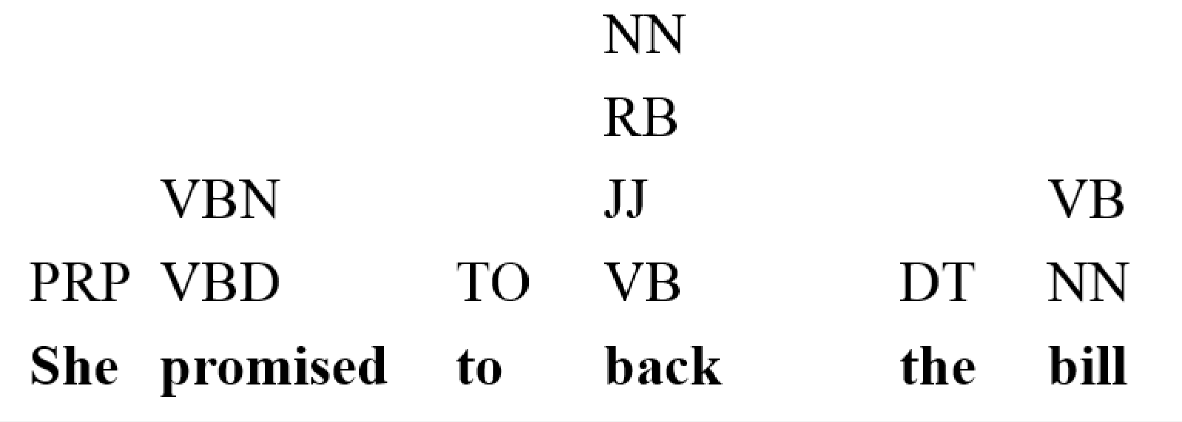
			NN		
			RB		
	VBN		JJ		VB
PRP	VBD	TO	VB	DT	NN
She	**promised**	**to**	**back**	**the**	**bill**

w_1^n = *she promised to back the bill*

$$t_1^n = \begin{cases} \text{PRP VBD TO VB DT NN} \\ \text{PRP VBN TO VB DT NN} \\ \text{PRP VBD TO JJ DT NN} \\ \ldots \end{cases}$$

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

Example



			NN		
			RB		
	VBN		JJ		VB
PRP	VBD	TO	VB	DT	NN
She	**promised**	**to**	**back**	**the**	**bill**

w_1^n = *she promised to back the bill*

$$t_1^n = \begin{cases} \text{PRP VBD TO VB DT NN} \\ \text{PRP VBN TO VB DT NN} \\ \text{PRP VBD TO JJ DT NN} \\ \ldots \end{cases}$$

the sequence with the largest probability given "*she promised to back the bill*"

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

is tough to calculate

- Why?

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

is tough to calculate

The probability of seeing *she promised to back the bill* in a corpus is small

Enter Bayes Rule

Bayes Rule allows us to break down any conditional probability into three components

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

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

So we can use Bayes Rule

- To decompose our HMM:

$$\hat{t}_1^n = \ argmax_{t_1^n}\, P(t_1^n|w_1^n)$$

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

So we can use Bayes Rule

To decompose our HMM:

$$\hat{t}_1^n = \ argmax_{t_1^n} \, P(t_1^n|w_1^n)$$

$$P(t_1^n|w_1^n) = \frac{P(w_1^n|t_1^n) \ * P(t_1^n)}{P(w_1^n)}$$

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

So we can use Bayes Rule

To decompose our HMM:

$$\hat{t}_1^n = \ argmax_{t_1^n} \, P(t_1^n|w_1^n)$$

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

Now before we go on

We can get rid of our denominator. Why?

$$\hat{t}_1^n = argmax_{t_1^n} \frac{P(w_1^n|t_1^n) \ * P(t_1^n)}{\boxed{P(w_1^n)}}$$

Now before we go on

We can get rid of our denominator. Why?

$$\hat{t}_1^n = argmax_{t_1^n} \frac{P(w_1^n|t_1^n) \; * P(t_1^n)}{\boxed{P(w_1^n)}}$$

$P(w_1^n)$ does not change for each possible tag sequence

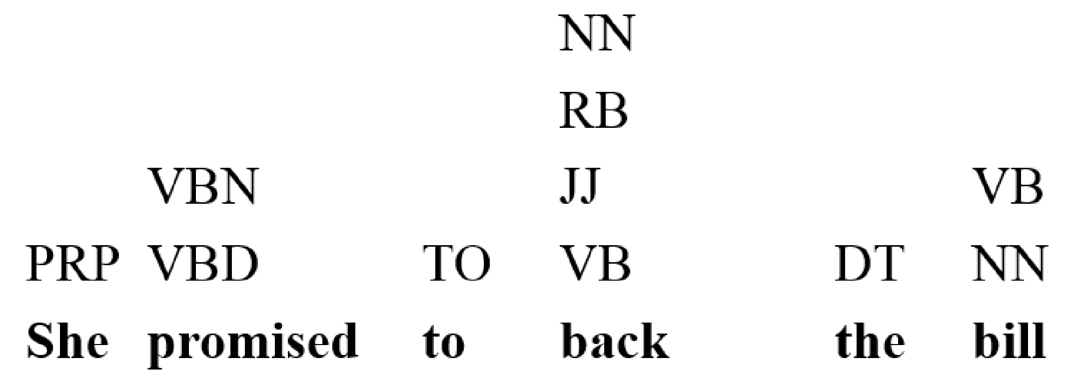
So we simplify our equation

$$\hat{t}_1^n = argmax_{t_1^n}\, P(w_1^n|t_1^n)\ * P(t_1^n)$$

Notation

$$\hat{t}_1^n = argmax_{t_1^n}\, \overset{\textit{\textbf{Likelihood}}}{P(w_1^n|t_1^n)} \, * \overset{\textit{\textbf{Prior}}}{P(t_1^n)}$$

A recap



			NN		
			RB		
	VBN		JJ		VB
PRP	VBD	TO	VB	DT	NN
She	**promised**	**to**	**back**	**the**	**bill**

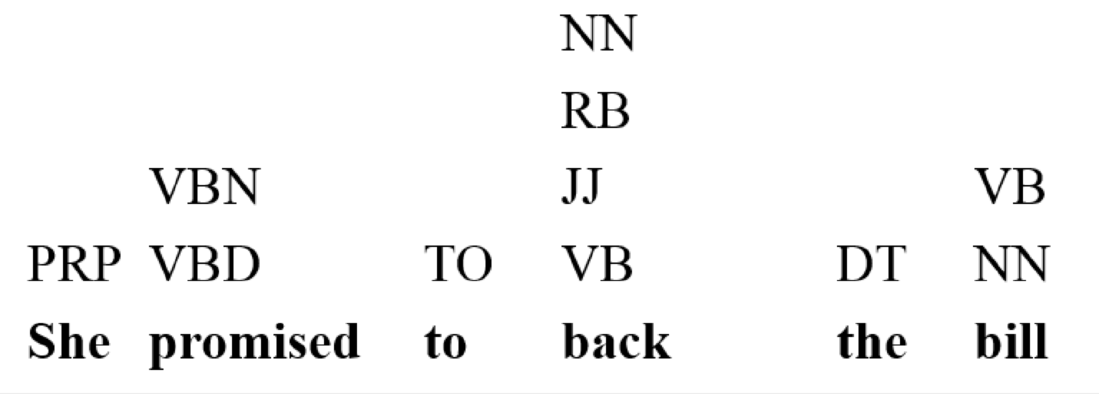
$$\hat{t}_1^n = argmax_{t_1^n} \boxed{P(w_1^n|t_1^n)} * P(t_1^n)$$

w_1^n = *she promised to back the bill*

probability of
"she promised to back the bill"
given a tag sequence

$$t_1^n = \begin{cases} \text{PRP VBD TO VB DT NN} \\ \text{PRP VBN TO VB DT NN} \\ \text{PRP VBD TO JJ DT NN} \\ \ldots \end{cases}$$

A recap



			NN		
			RB		
	VBN		JJ		VB
PRP	VBD	TO	VB	DT	NN
She	**promised**	**to**	**back**	**the**	**bill**

$$\hat{t}_1^n \approx argmax_{t_1^n} P(w_1^n | t_1^n) \boxed{* P(t_1^n)}$$

probability of the tag sequence

w_1^n = *she promised to back the bill*

t_1^n =
- PRP VBD TO VB DT NN
- PRP VBN TO VB DT NN
- PRP VBD TO JJ DT NN
- …

Let’s look at our likelihood

$$P(w_1^n|t_1^n)$$

Still difficult to calculate: why?

Let’s look at our likelihood

$$P(w_1^n|t_1^n)$$

Still difficult to calculate: why?

The probability of seeing *she promised to back the bill* in a corpus is small

Let’s look at our likelihood

$$P(w_1^n|t_1^n)$$

Still difficult to calculate: why?

The probability of seeing *she promised to back the bill* in a corpus is small

So what do we do?

Two Assumptions

- Assumption 1
- Assumption 2

Assumption 1

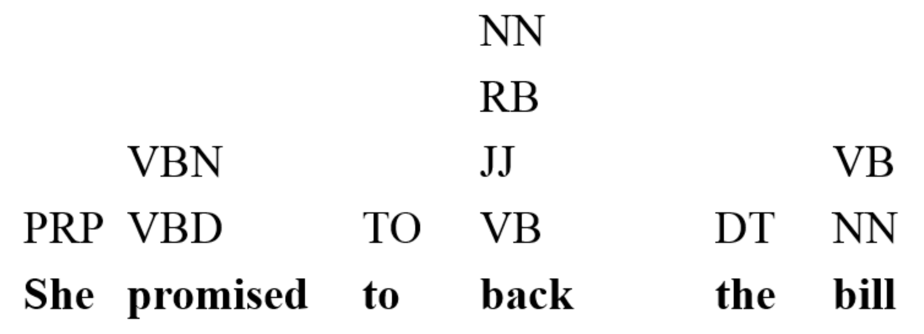
- the probability of a word appearing depends only on its own POS tag
- It is independent of the other words and tags around it

Assumption 1

- the probability of a word appearing depends only on its own POS tag
- It is independent of the other words and tags around it

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

Assumption 1



			NN		
			RB		
	VBN		JJ		VB
PRP	VBD	TO	VB	DT	NN
She	**promised**	**to**	**back**	**the**	**bill**

- the probability of a word appearing depends only on its own POS tag
- It is independent of the other words and tags around it

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

$$P(she\ promised\ to\ back\ the\ bill\ | PRP\ VBD\ TO\ VB\ DT\ NN)\ \approx$$

$$P(she|PRP) * P(promised|VBD) * P(to|TO) * P(back|VB) * P(the|DT) * P(bill|NN)$$

Assumption 1 plugged in

$$\hat{t}_1^n = argmax_{t_1^n}\, P(w_1^n | t_1^n) \; * P(t_1^n)$$

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

Assumption 1 plugged in

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

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

Assumption 2

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

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

Assumption 2

- the probability of a tag appearing depends only on the previous tag

Remember our Markov assumption

- Estimate the conditional probability of the next word without looking too far in the past

$$P(w_n | w_1^{n-1}) \approx P(w_n | w_{n-N+1}^{n-1})$$

Assumption 2

- the probability of a tag appearing depends only on the previous tag

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

What model is this?

Assumption 2

- the probability of a tag appearing depends only on the previous tag

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

What model is this? Bigram model

Assumption 2 plugged in

$$\hat{t}_1^n \quad \approx \; argmax_{t_1^n} \prod_{i=1}^{n} P(w_i|t_i) * P(t_1^n)$$

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

How do we calculate the individual probabilities

$$\hat{t}_1^n \approx argmax_{t_1^n} \prod_{i=1}^{n} P(w_i|t_i) * P(t_i|t_{i-1})$$

$P(w_i|t_i)$ = ?

Remember from your relative frequency tables?

$$\hat{t}_1^n \approx argmax_{t_1^n} \prod_{i=1}^{n} P(w_i|t_i) * P(t_i|t_{i-1})$$

How do we calculate the individual probabilities

$$\hat{t}_1^n \approx argmax_{t_1^n} \prod_{i=1}^{n} P(w_i|t_i) * P(t_i|t_{i-1})$$

How do we calculate the individual probabilities

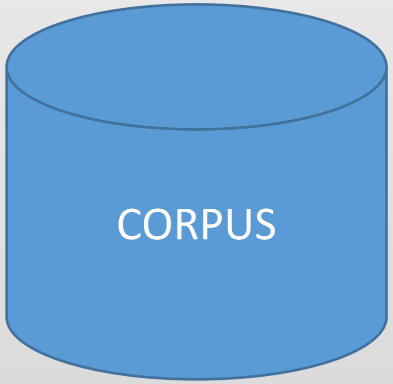
$$\hat{t}_1^n \approx argmax_{t_1^n} \prod_{i=1}^{n} \boxed{P(w_i|t_i)} * P(t_i|t_{i-1})$$

How do we calculate the individual probabilities

$$\hat{t}_1^n \approx argmax_{t_1^n} \prod_{i=1}^{n} \boxed{P(w_i|t_i)} * P(t_i|t_{i-1})$$

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

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



	she	promised	to	back	the	bill
PRP	58	8	0	0	0	0
VBN	2	24	0	1	0	4
VBD	2	34	4	1	0	9
TO	0	0	68	0	1	2
NN	1	0	0	43	0	82
RB	1	0	15	0	1	1
JJ	2	0	0	32	0	0
VB	1	10	1	60	0	40
DT	4	0	3	0	115	0

PRP	VBN	VBD	TO	NN	RB	JJ	VB	DT
2533	927	2417	746	158	1093	341	278	3000

$$P(promised|VBD) = ?$$

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

CORPUS

	she	promised	to	back	the	bill
PRP	58	8	0	0	0	0
VBN	2	24	0	1	0	4
VBD	2	34	4	1	0	9
TO	0	0	68	0	1	2
NN	1	0	0	43	0	82
RB	1	0	15	0	1	1
JJ	2	0	0	32	0	0
VB	1	10	1	60	0	40
DT	4	0	3	0	115	0

PRP	VBN	VBD	TO	NN	RB	JJ	VB	DT
2533	927	2417	746	158	1093	341	278	3000

$$P(promised|VBD) =$$

$$\frac{frequency(promised, VBD)}{frequency(VBD)}$$

$$= \frac{34}{2417} = 0.14$$

How do we calculate the individual probabilities

$$\hat{t}_1^n \approx argmax_{t_1^n} \prod_{i=1}^{n} P(w_i|t_i) * P(t_i|t_{i-1})$$

How do we calculate the individual probabilities

$$\hat{t}_1^n \approx argmax_{t_1^n} \prod_{i=1}^{n} P(w_i|t_i) * \boxed{P(t_i|t_{i-1})}$$

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

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

CORPUS

t_i

t_{i-1}	PRP	VBN	VBD	TO	NN	RB	JJ	VB	DT
PRP	0	30	80	0	0	0	0	0	0
VBN	2	0	0	1	0	4	8	14	34
VBD	2	34	0	1	0	9	9	9	45
TO	0	0	68	0	1	2	45	2	28
NN	1	0	0	43	60	82	8	2	23
RB	1	0	15	0	1	0	0	0	0
JJ	2	0	0	32	0	0	23	0	14
VB	1	10	1	60	0	40	4	0	12
DT	14	0	13	0	115	2	14	10	0

PRP	VBN	VBD	TO	NN	RB	JJ	VB	DT
2533	927	2417	746	158	1093	341	278	3000

$$P(VBD|PRP) = ?$$

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

CORPUS

t_i

t_{i-1}

	PRP	VBN	VBD	TO	NN	RB	JJ	VB	DT
PRP	0	30	80	0	0	0	0	0	0
VBN	2	0	0	1	0	4	8	14	34
VBD	2	34	0	1	0	9	9	9	45
TO	0	0	68	0	1	2	45	2	28
NN	1	0	0	43	60	82	8	2	23
RB	1	0	15	0	1	0	0	0	0
JJ	2	0	0	32	0	0	23	0	14
VB	1	10	1	60	0	40	4	0	12
DT	14	0	13	0	115	2	14	10	0

PRP	VBN	VBD	TO	NN	RB	JJ	VB	DT
2533	927	2417	746	158	1093	341	278	3000

$$P(VBD|PRP) =$$

$$\frac{frequency(PRP,VBD)}{frequency(PRP)}$$

$$= \frac{80}{2533} = 0.03$$

$$\hat{t}_1^n = \ argmax_{t_1^n} \, P(t_1^n|w_1^n)$$

We went from here to here in a number of steps

So lets recap…

$$\hat{t}_1^n \approx argmax_{t_1^n} \prod_{i=1}^{n} P(w_i|t_i) * P(t_i|t_{i-1})$$

$$\hat{t}_1^n = \ argmax_{t_1^n} \, P(t_1^n | w_1^n)$$

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

Bayes Rule

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

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

Bayes Rule

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

Removed denominator because same over all calculations – doesn’t change the outcome

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

$$\hat{t}_1^n = \ argmax_{t_1^n} \, P(t_1^n | w_1^n)$$

Bayes Rule

$$\hat{t}_1^n = argmax_{t_1^n} \, \frac{P(w_1^n | t_1^n) \ * P(t_1^n)}{P(w_1^n)}$$

Removed denominator because same over all calculations – doesn’t change the outcome

$$\hat{t}_1^n = argmax_{t_1^n} \, P(w_1^n | t_1^n) \ * P(t_1^n)$$

Assumption 1:
words are independent of their tags

$$\hat{t}_1^n \approx \ argmax_{t_1^n} \prod_{i=1}^{n} P(w_i | t_i) * P(t_1^n)$$

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

Bayes Rule

$$\hat{t}_1^n = argmax_{t_1^n} \frac{P(w_1^n | t_1^n) \; * P(t_1^n)}{P(w_1^n)}$$

Removed denominator
because same over all
calculations – doesn't
change the outcome

$$\hat{t}_1^n = argmax_{t_1^n} P(w_1^n | t_1^n) \; * P(t_1^n)$$

Assumption 1:
words are independent
of their tags

$$\hat{t}_1^n \approx argmax_{t_1^n} \prod_{i=1}^{n} P(w_i | t_i) * P(t_1^n)$$

Assumption 2:
Markov assumption
Bigram Model

$$\hat{t}_1^n \approx argmax_{t_1^n} \prod_{i=1}^{n} P(w_i | t_i) * P(t_i | t_{i-1})$$

we have our equation!

$$\hat{t}_1^n \approx argmax_{t_1^n} \prod_{i=1}^{n} P(w_i|t_i) * P(t_i|t_{i-1})$$

we have our equation!

$$\hat{t}_1^n \approx argmax_{t_1^n} \prod_{i=1}^{n} P(w_i|t_i) * P(t_i|t_{i-1})$$

which we know is just a for loop
over each possible tag sequence

we have our equation!

$$\hat{t}_1^n \approx argmax_{t_1^n} \prod_{i=1}^{n} P(w_i|t_i) * P(t_i|t_{i-1})$$

which we know is just a for loop over each possible tag sequence

and we are returning the tag sequence with the greatest probability

$$\hat{t}_1^n \approx argmax_{t_1^n} \prod_{i=1}^{n} P(w_i|t_i) * P(t_i|t_{i-1})$$

	PRP	VBN	VBD	TO	NN	RB	JJ	VB	DT
PRP	0	8	0	0	0	0	0	0	0
VBN	2	0	0	1	0	4	8	14	34
VBD	2	34	0	1	0	9	9	9	45
TO	0	0	68	0	1	2	45	2	28
NN	1	0	0	43	60	82	8	2	23
RB	1	0	15	0	1	0	0	0	0
JJ	2	0	0	32	0	0	23	0	14
VB	1	10	1	60	0	40	4	0	12
DT	14	0	13	0	115	2	14	10	0

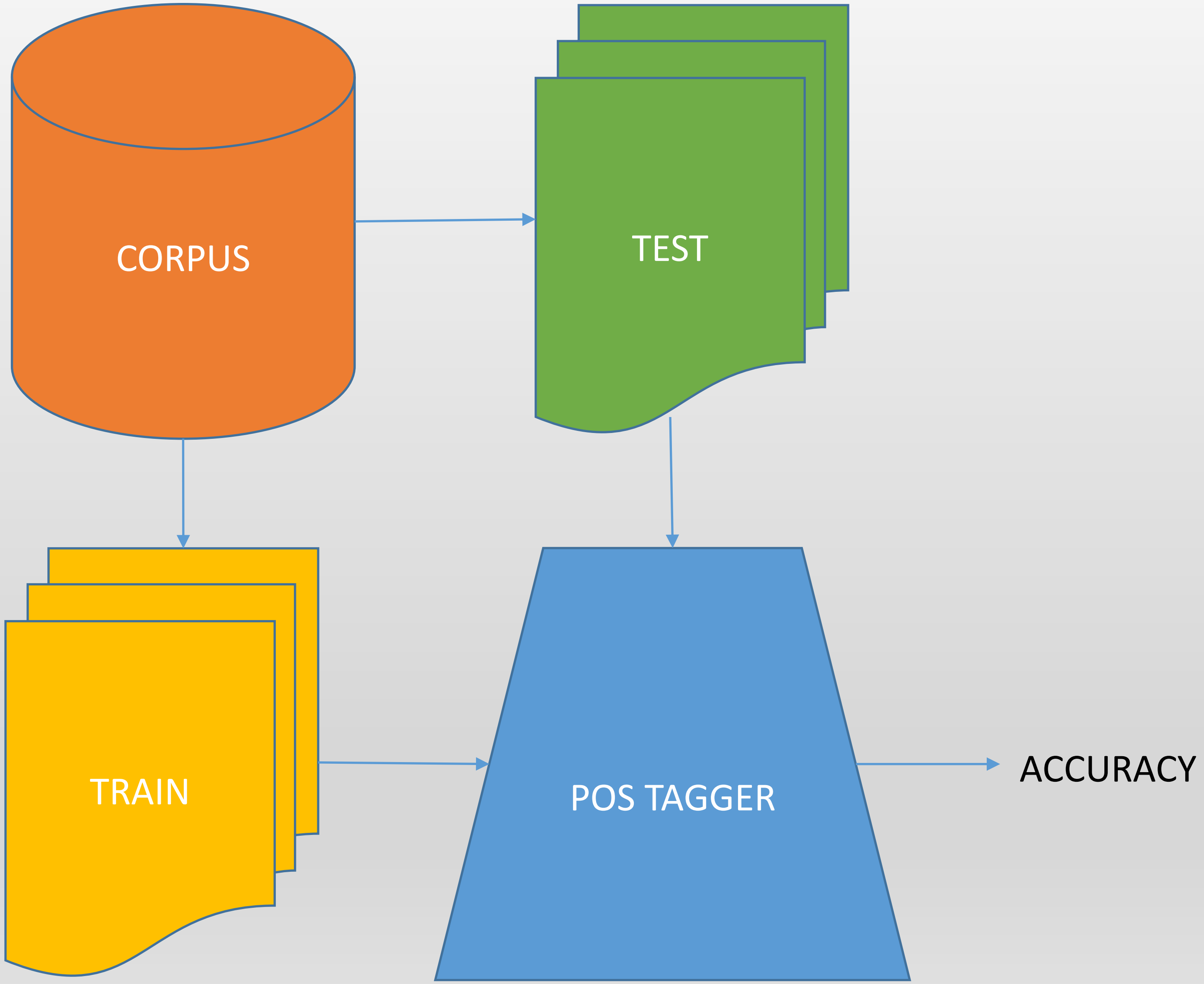
	she	promised	to	back	the	bill
PRP	58	8	0	0	0	0
VBN	2	24	0	1	0	4
VBD	2	34	4	1	0	9
TO	0	0	68	0	1	2
NN	1	0	0	43	0	82
RB	1	0	15	0	1	1
JJ	2	0	0	32	0	0
VB	1	10	1	60	0	40
DT	4	0	3	0	115	0

PRP	VBN	VBD	TO	NN	RB	JJ	VB	DT
2533	927	2417	746	158	1093	341	278	3000

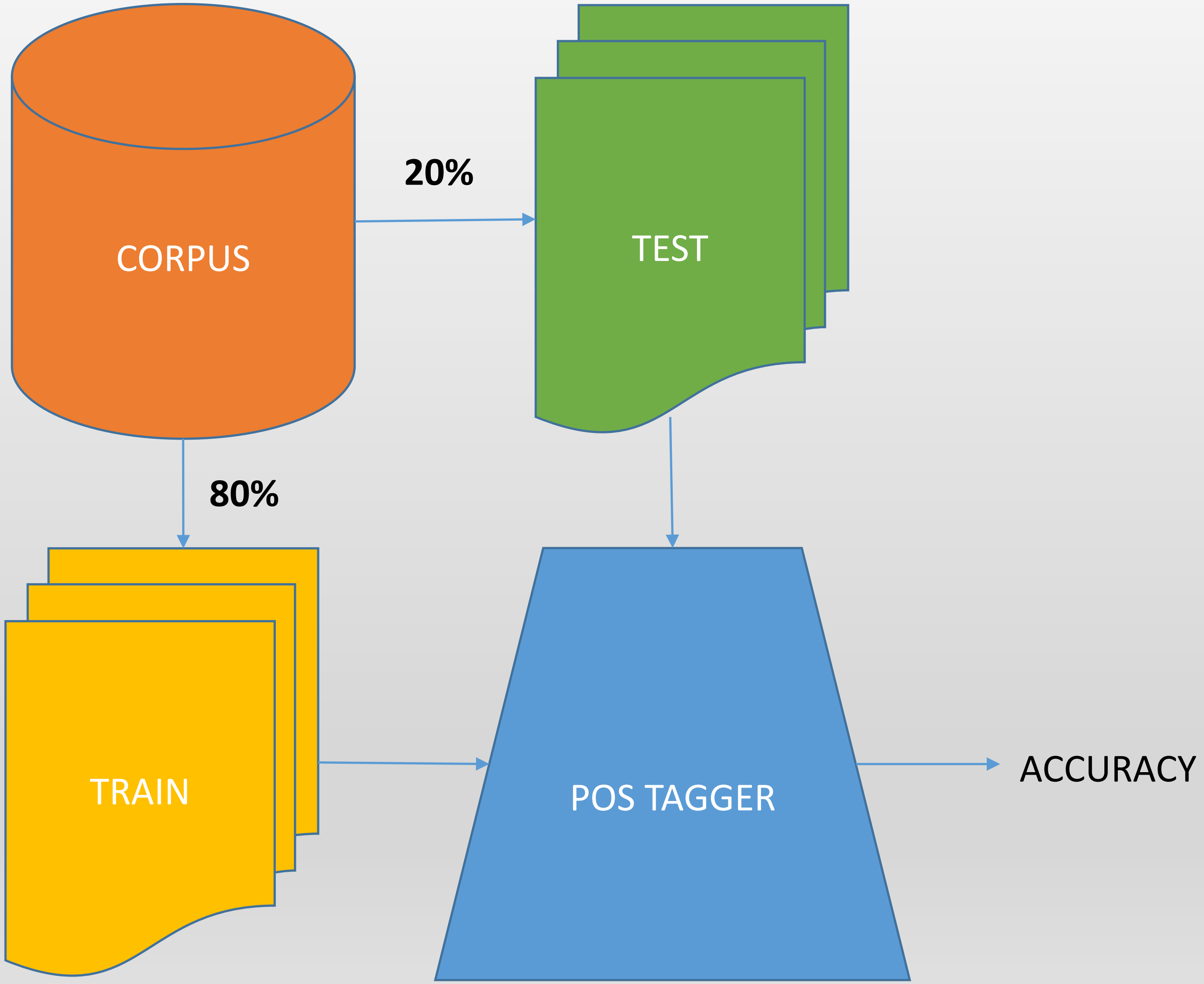
Evaluation Methodology

- Train/Test set
- Dev/Train/Test set
- k-Fold Cross Validation

Evaluation



Evaluation:
80-20



CORPUS

20%

TEST

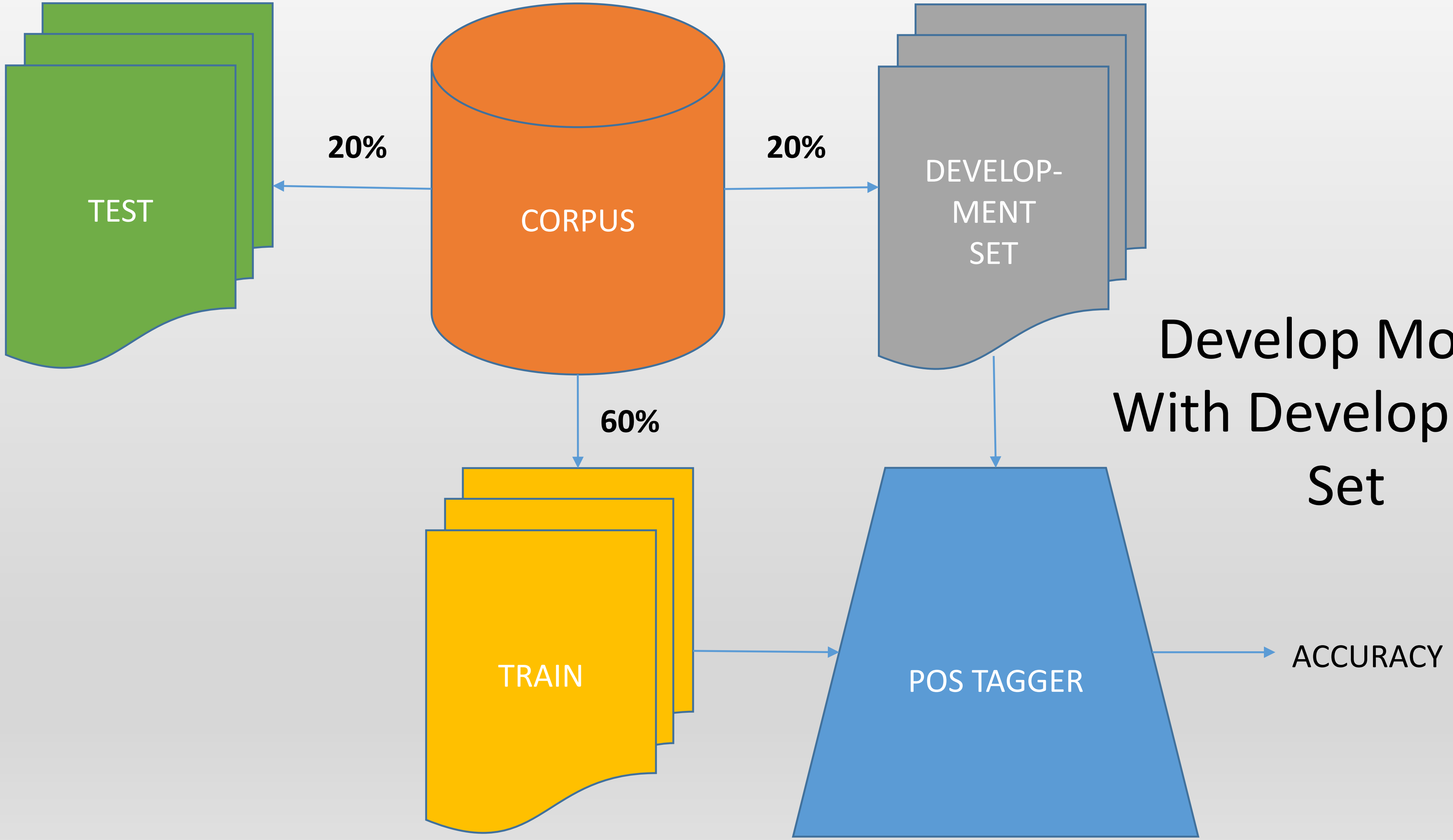
80%

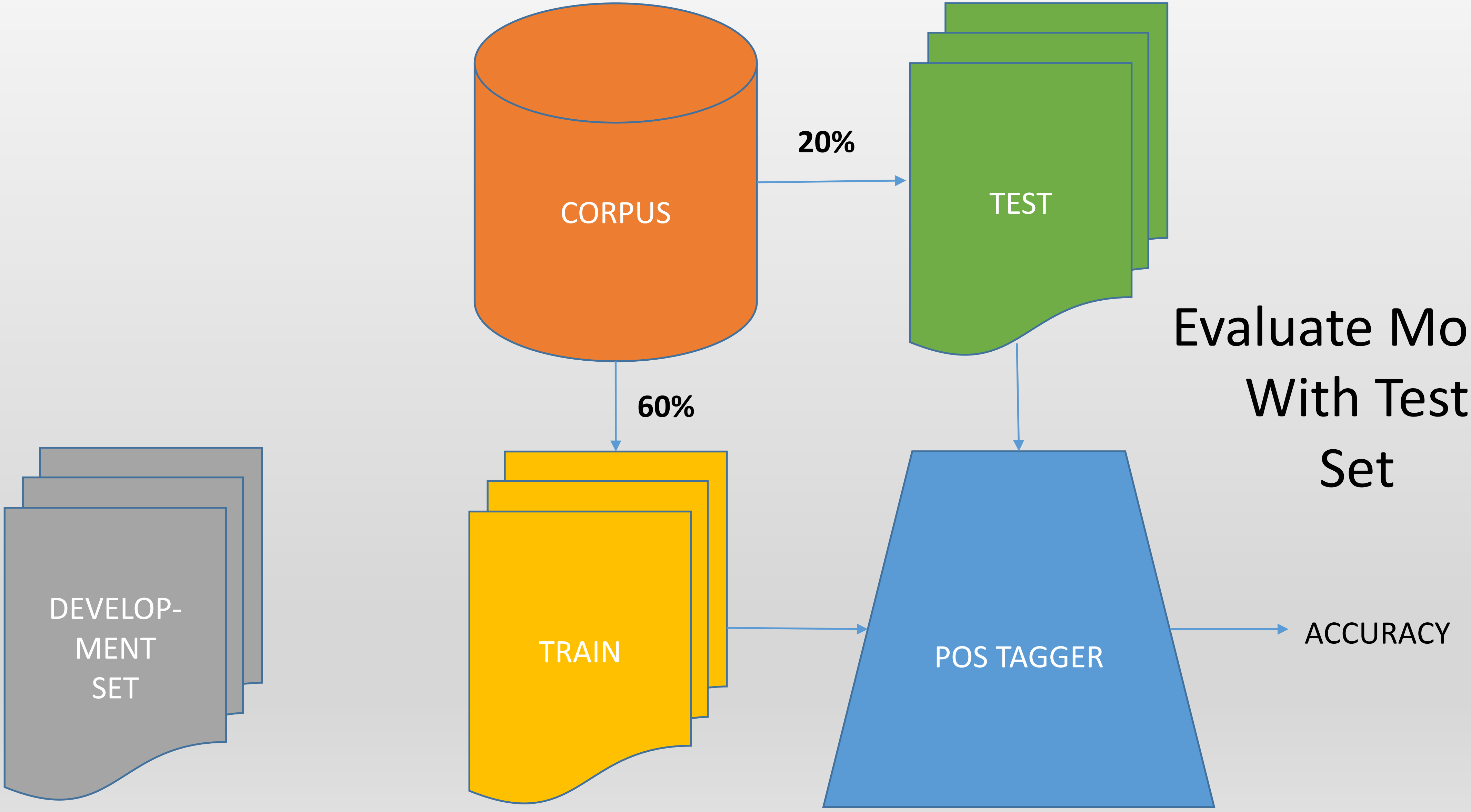
TRAIN

POS TAGGER

ACCURACY

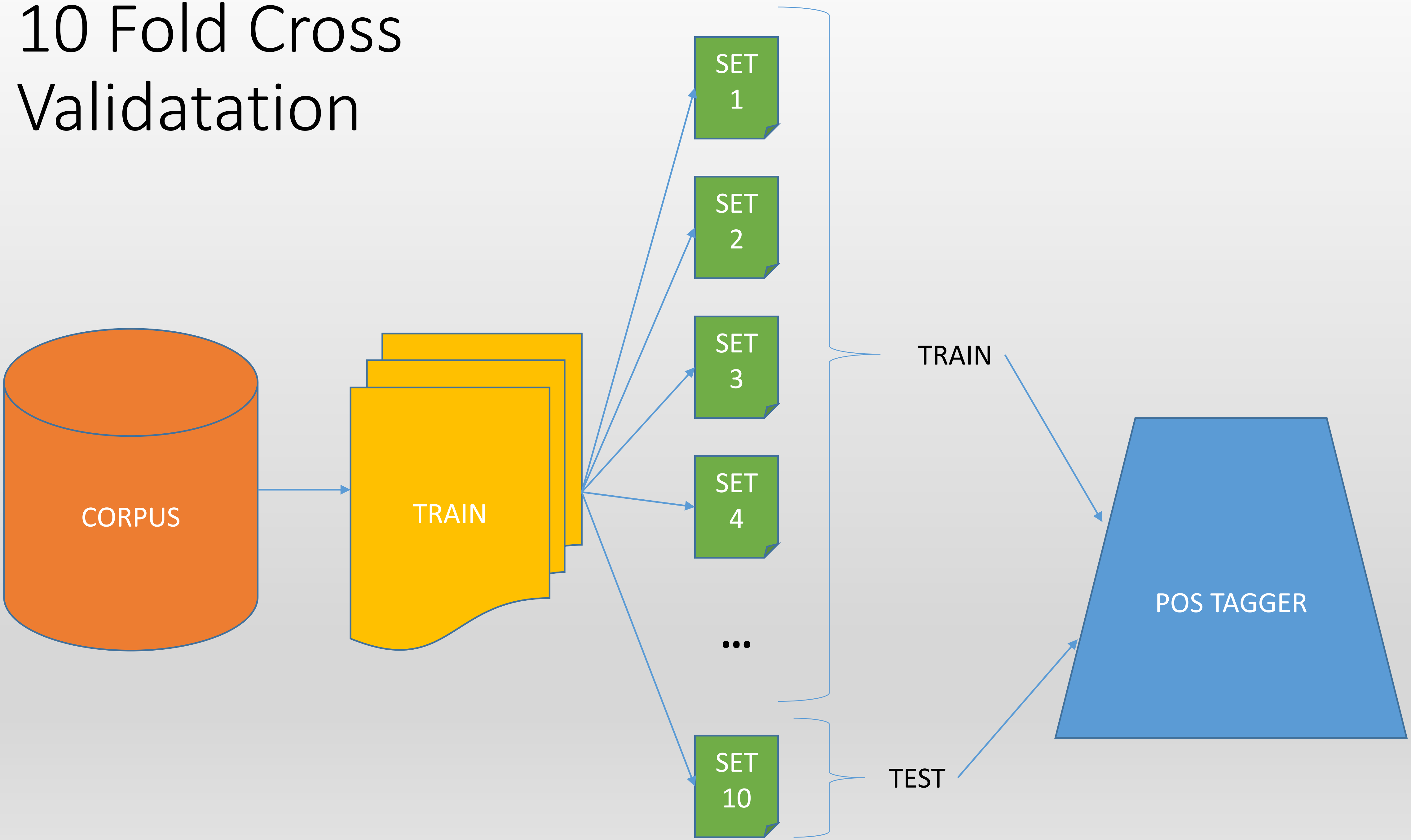
Develop Model With Development Set





Evaluate Model With Test Set

10 Fold Cross Validatation



CORPUS

TRAIN

SET 1

SET 2

SET 3

SET 4

...

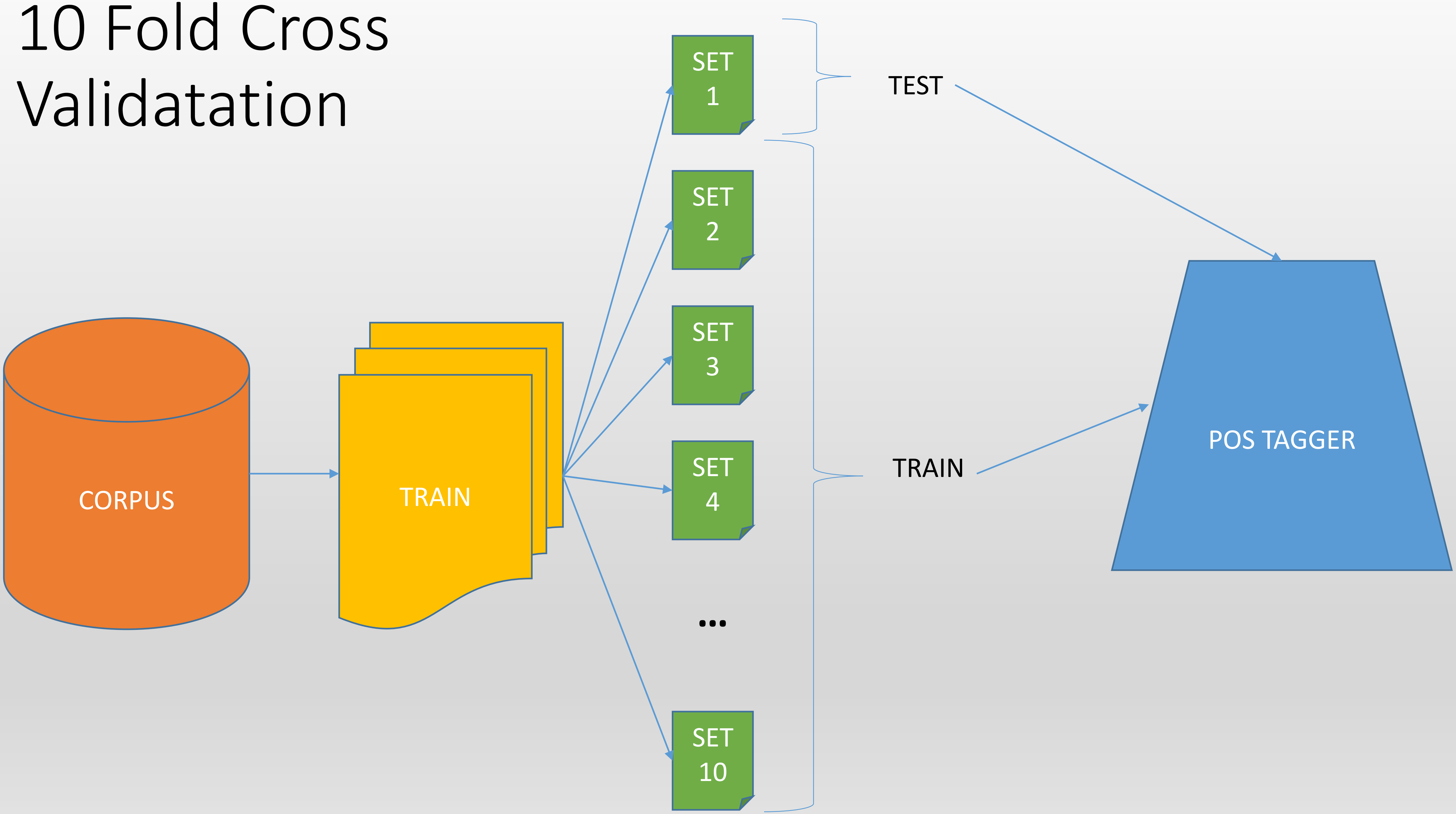
SET 10

TRAIN

TEST

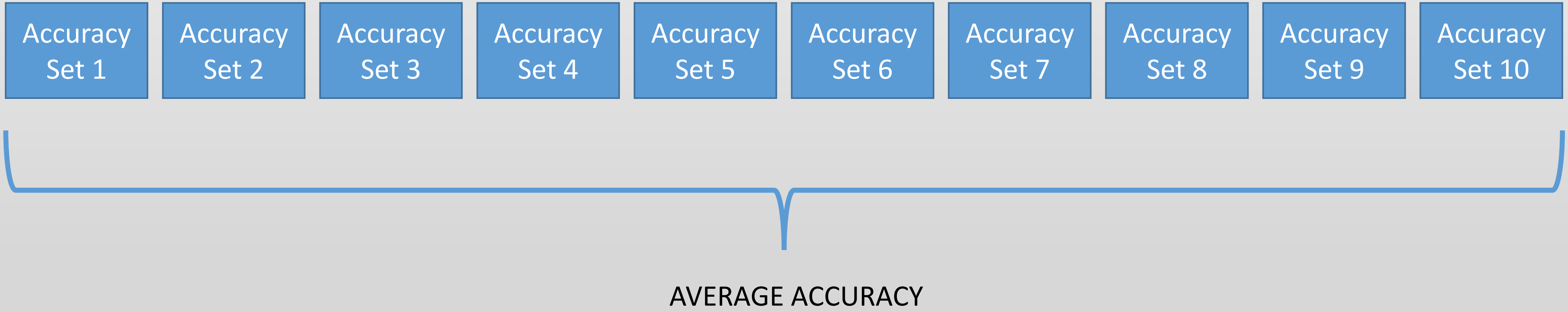
POS TAGGER

10 Fold Cross Validatation



Cycle Through

- Each file is used as a test once and only once
    - With the remaining used as train



Accuracy Set 1

Accuracy Set 2

Accuracy Set 3

Accuracy Set 4

Accuracy Set 5

Accuracy Set 6

Accuracy Set 7

Accuracy Set 8

Accuracy Set 9

Accuracy Set 10

AVERAGE ACCURACY

Accuracy

statistical measure of how well

The classification went

- $accuracy = \frac{number\ correctly\ tagged\ words}{total\ number\ of\ words}$

How good is your algorithm?

- Gold standard
  - Human labeled gold standard

- Baseline comparison
  - Most frequent class baseline: assign each token to the most frequent class in the training set – in this case the most frequent POS

Error Analysis

- Confusion matrix:
  - E.g. which tags did we most often confuse with which other tags?
  - How much of the overall error does each confusion account for?

PREDICTED CLASSES

ACTUAL CLASSES

	VB	TO	NN
VB			
TO			
NN			

Error Analysis

PERFECT CLASSIFIER would have zeros in all cells except for the diagonal

Simple example, with ten instances of each POS Tag

PREDICTED CLASSES

ACTUAL CLASSES

	VB	TO	NN
VB	**10**	0	0
TO	0	**10**	0
NN	0	0	**10**

Error Analysis

take a look at the confusion matrix, see where things are going wrong and if there is a rule that you could incorporate to aid in fixing this

PREDICTED CLASSES

ACTUAL CLASSES

	VB	TO	NN
VB	**5**	0	5
TO	0	**10**	0
NN	10	0	**0**

Questions?

Programming Assignment #3

- POS Tagger (tagger.py)
  - Simple baseline tagger maximizes P(tag|word)
  - 5 additional rules
- Scoring program (scorer.py)
  - Calculates the overall accuracy of the tagger
  - Calculates the confusion matrix

- PA3.zip
  - pos-train.txt
  - pos-test.txt
  - pos-test-key.txt

PA3.zip

- pos-test.txt

```
[ 61/CD years/NNS ]
old/JJ ,/, will/MD join/VB
[ the/DT board/NN ]
as/IN
[ a/DT nonexecutive/JJ director/NN Nov./NNP 29/CD ]
./.
```

- pos-test.txt

```
No ,
[ it ]
[ was n't Black Monday ]
.
```

- pos-test-key.txt

```
No/RB ,/,
[ it/PRP ]
[ was/VBD n't/RB Black/NNP Monday/NNP ]
./.
```

Programs

- To run the tagger:

```
python tagger.py pos-train.txt pos-test.txt > pos-test-with-tags.txt
```

- To run the scorer:

```
python scorer.py pos-test-with-tags.txt pos-test-key.txt > pos-tagging-report.txt
```