Natural Language Processing

Paper Code: CS-821/6
UG 8th Sem

Dr. Samit Biswas, *Assistant Professor*

Department of Computer Sc. and Technology

Indian Institute of Engineering Science and Technology, Shibpur

Email: samit@cs.iiests.ac.in

Plan for Today

- Part of Speech Tagging

Part-of-Speech Tagging

- PoS Tagging is the Process of making up a word in a corpus to a corresponding part of speech tag based on its context and definition.

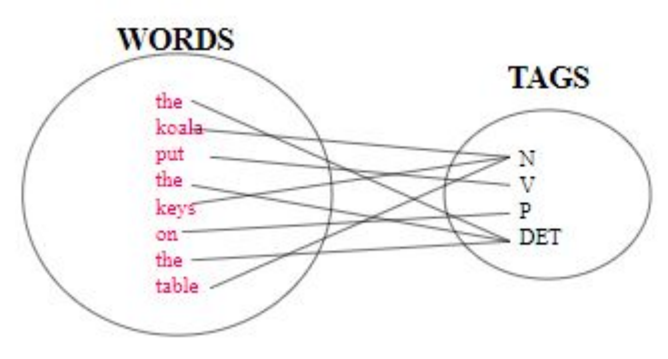
Input: the lead paint is unsafe

Output: the/Det lead/N paint/N is/V unsafe/Adj

How many Part-of-Speech are there in English?

- **Open classes:**
  - *nouns, verbs, adjectives, adverbs*
- **Closed classes:** function words
  - *conjunctions*: *and, or, but*
  - *pronounts*: *I, she, him*
  - *prepositions*: *with, on*
  - *determiners*: *the, a, an*

- The process of assigning a part-of-speech or lexical class marker to each word in a corpus:



Application of PoS Tagging

- Speech synthesis pronunciation
- Parsing: e.g. *Time flies like an arrow*
  - Is *flies* an N or V?
- Word prediction in speech recognition
  - Possessive pronouns (*my, your, her*) are likely to be followed by nouns
  - Personal pronouns (*I, you, he*) are likely to be followed by verbs
- Machine Translation

Choosing a POS Tagset

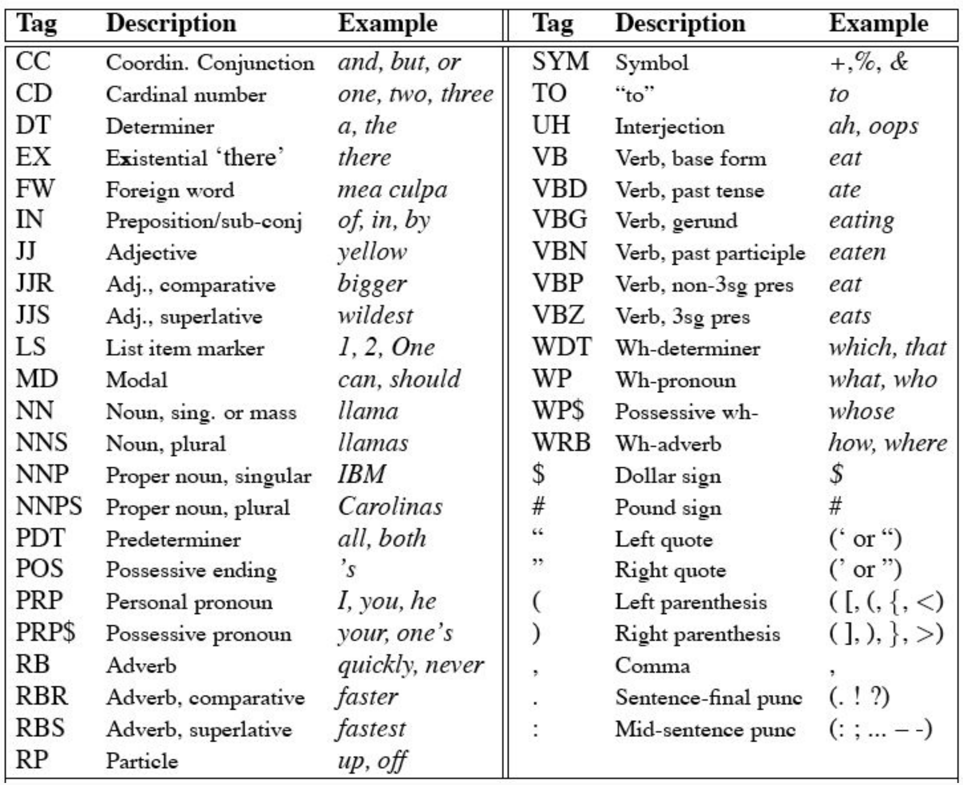
Some different tag sets have been proposed for PoS tagging

- Brown corpus tagset (87 tags):
- Penn Treebank tagset (45 tags):
- C7 tagset (146 tags)

Choosing a POS Tagset

- To do POS tagging, first need to choose a set of tags
- Could pick very small tagsets
  - N, V, Adj, Adv.
- 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

- 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*			

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***

Algorithms for POS Tagging - Approaches

- Basic approaches
    - Rule-Based
    - Transformation-based tagging
        - Learned rules (statistics and linguistic)
        - E.g., Brill tagger
    - Probabilistic Tagging
        - HMM (Hidden Markov Model) tagging

Rule Based Tagging

- Typically…start with a dictionary of words and possible tags
- Assign all possible tags to words using the dictionary
- Write rules by hand to *selectively remove* tags
- Stop when each word has exactly one (presumably correct) tag

Rule Based Tagging ...

Start with a POS Dictionary

- she: PRP
- promised: VBN,VBD
- to: TO
- back: VB, JJ, RB, NN
- the: DT
- bill: NN, VB
- etc,… for almost all words of English

Rule Based Tagging ...

Assign All Possible POS to Each Word

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

Rule Base Tagging ...

Apply rules eliminating some PoS

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

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

Transformation based (Brill) Tagging

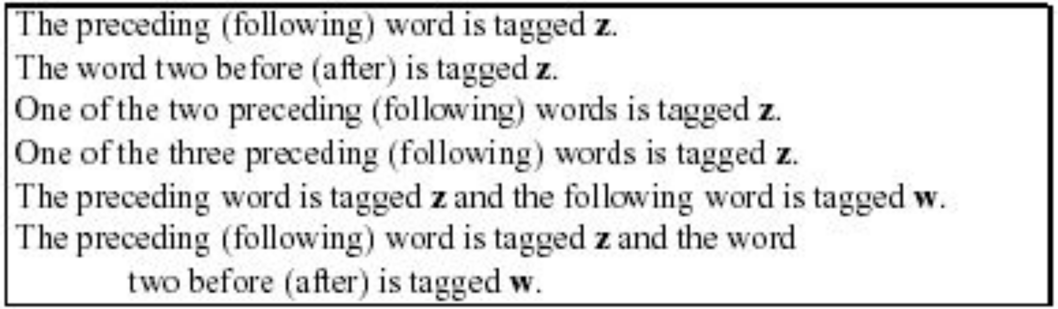
- Combines Rule-based and Stochastic Tagging
  - Like rule-based because rules are used to specify tags in a certain environment
  - Like stochastic approach because we use a tagged corpus to find the best performing rules
    - *Rules are learned from data*
- Input:
  - Tagged corpus
  - Dictionary (*with most frequent tags*)

Transformation based (Brill) Tagging

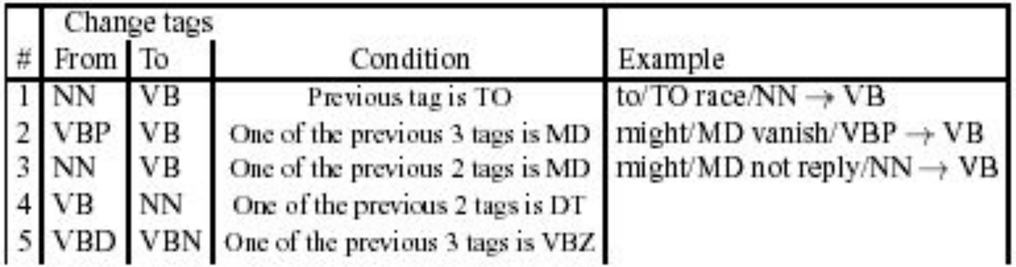
Transformation-Based Tagging

- Basic Idea: Strip tags from tagged corpus and try to learn them by rule application
  - For untagged, first initialize with most probable tag for each word
  - Change tags according to best rewrite rule, e.g. *“if word-1 is a determiner and word-2 is a verb then change the tag to noun”*
  - Compare to gold standard
  - Iterate
- Rules created via rule templates, e.g.of the form *if word-1 is an X and word-2 is a Y then change the tag to Z”*
  - Find rule that applies correctly to most tags and apply
  - Iterate on newly tagged corpus until threshold reached
  - Return ordered set of rules
- NB: Rules may make errors that are corrected by later rules

Templates for TBL



The preceding (following) word is tagged **z**.
The word two before (after) is tagged **z**.
One of the two preceding (following) words is tagged **z**.
One of the three preceding (following) words is tagged **z**.
The preceding word is tagged **z** and the following word is tagged **w**.
The preceding (following) word is tagged **z** and the word
two before (after) is tagged **w**.



	Change tags			
#	From	To	Condition	Example
1	NN	VB	Previous tag is TO	to/TO race/NN → VB
2	VBP	VB	One of the previous 3 tags is MD	might/MD vanish/VBP → VB
3	NN	VB	One of the previous 2 tags is MD	might/MD not reply/NN → VB
4	VB	NN	One of the previous 2 tags is DT	
5	VBD	VBN	One of the previous 3 tags is VBZ	

Sample TBL Rule Application

- Labels every word with its most-likely tag
  - E.g. *race* occurences in the Brown corpus:
    - $P(NN|race) = .98$
    - $P(VB|race)= .02$
    - *is/VBZ expected/VBN to/TO race/NN tomorrow/NN*
- Then TBL applies the following rule
  - "Change NN to VB when previous tag is TO"

    … *is/VBZ expected/VBN to/TO race/NN tomorrow/NN*

    becomes

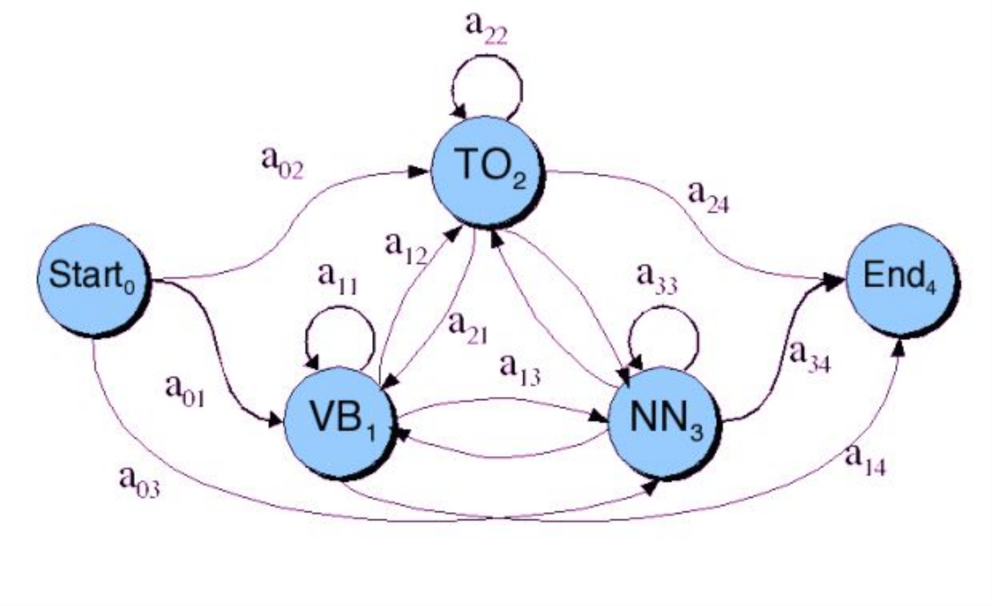
    … *is/VBZ expected/VBN to/TO race/VB tomorrow/NN*

TBL Tagging Algorithm

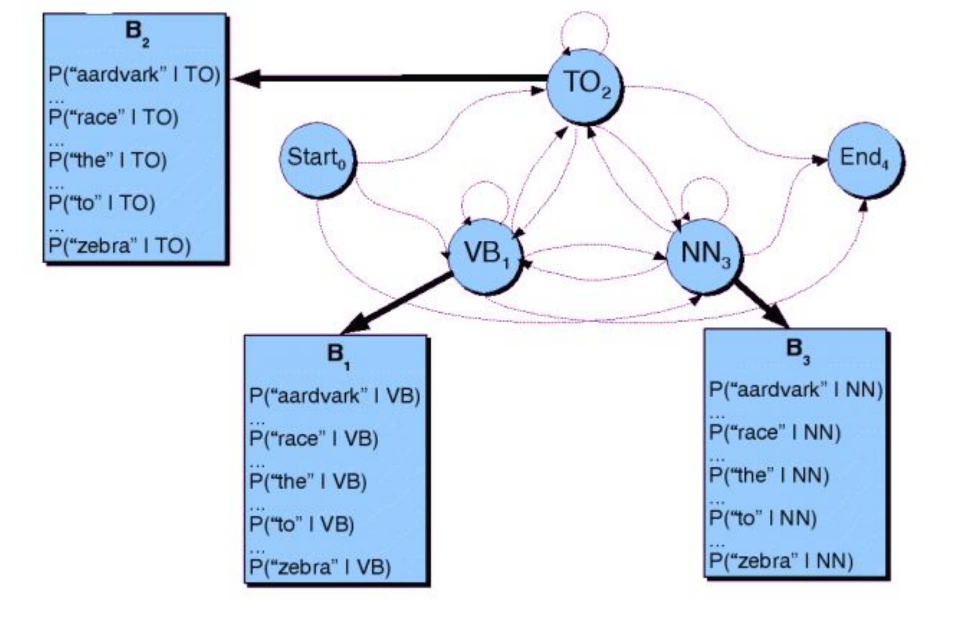
- **Step 1:** Label every word with most likely tag (from dictionary)
- **Step 2:** Check every possible transformation & select one which most improves tag accuracy.
- **Step 3:** Re-tag corpus applying this rule, and add rule to end of rule set
- **Repeat 2-3** until some stopping criterion is reached, e.g., X% correct with respect to training corpus
- **RESULT:** Ordered set of transformation rules to use on new data tagged only with most likely POS tags

HMM based Tagging

- The HMM hidden states, the POS tags, can be represented in a graph where the edges are the transition probabilities between POS tags.



- Word likelihoods for POS HMM
- For each POS tag, give words with probabilities

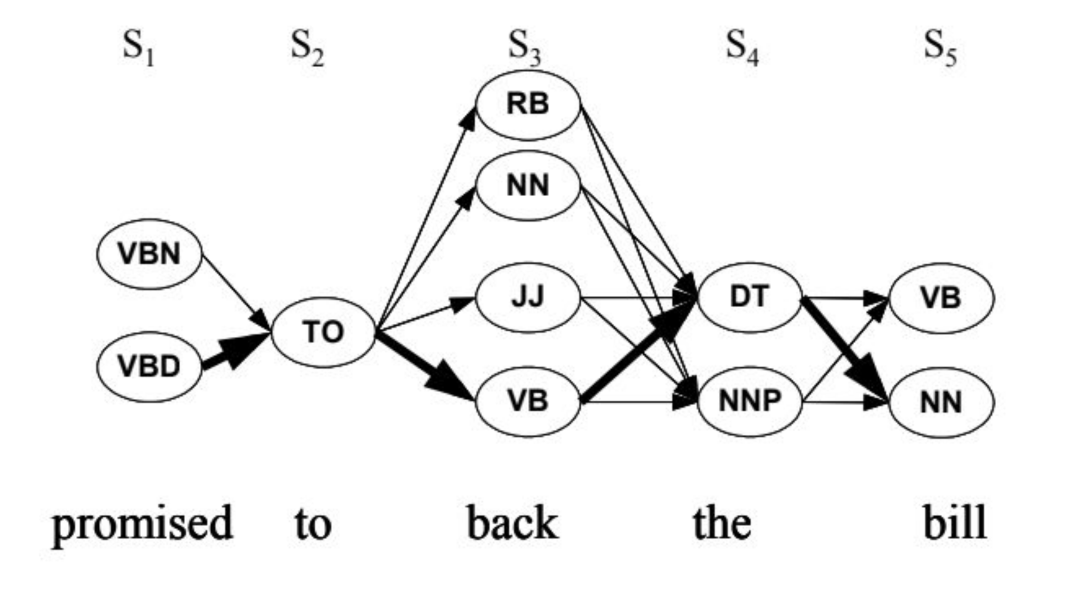


Using HMMs for POS tagging

- From the tagged corpus, create a tagger by computing the two matrices of probabilities, A and B
  - Straightforward for bigram HMM, done by counting
  - For higher-order HMMs, efficiently compute matrix by the forward-backward algorithm
- To apply the HMM tagger to unseen text, we must find the best sequence of transitions
  - Given a sequence of words, find the sequence of states (POS tags) with the highest probabilities along the path
  - This task is sometimes called “decoding”
  - Use the Viterbi algorithm

Viterbi intuition: we are looking for the best ‘path’

Each word has states representing the possible POS tags:



References

References

- Daniel Jurafsky & James H. Martin, “*Speech and Language Processing*”, Prentice Hall
- Diana Maynard, Kalina Bontcheva, Isabelle Augenstein, “*Natural Language Processing for the Semantic Web*”, A Publication in the Morgan & Claypool Publishers series.
- Steven Bird, Ewan Klein and Edward Loper, “*Natural Language Processing with Python*”, By O‘Reilly.
- Christopher Manning and Hinrich Schtze, “*Foundations of Statistical Natural Language Processing*”, The MIT Press.

Thank You

Contacts:

Dr. Samit Biswas
Department of CST,
IIEST, Shibpur

samit@cs.iiests.ac.in
https://www.iiests.ac.in/IIEST/Faculty/cs-samit