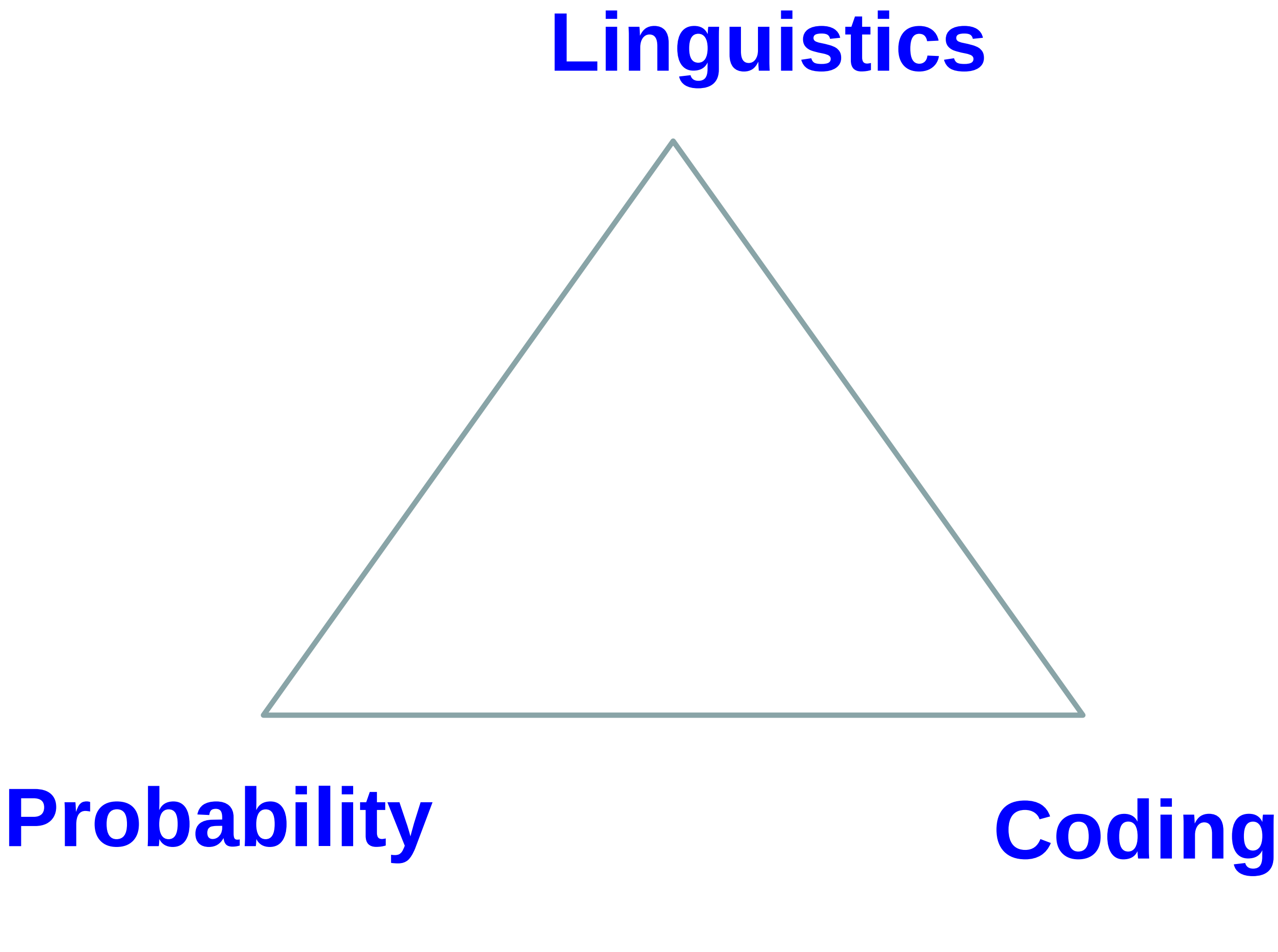
CS626: Speech, Natural Language Processing and the Web

Part of Speech Tagging

Pushpak Bhattacharyya
Computer Science and Engineering Department
IIT Bombay

Week 3 of 8th August, 2022

The Trinity



NLP Layers

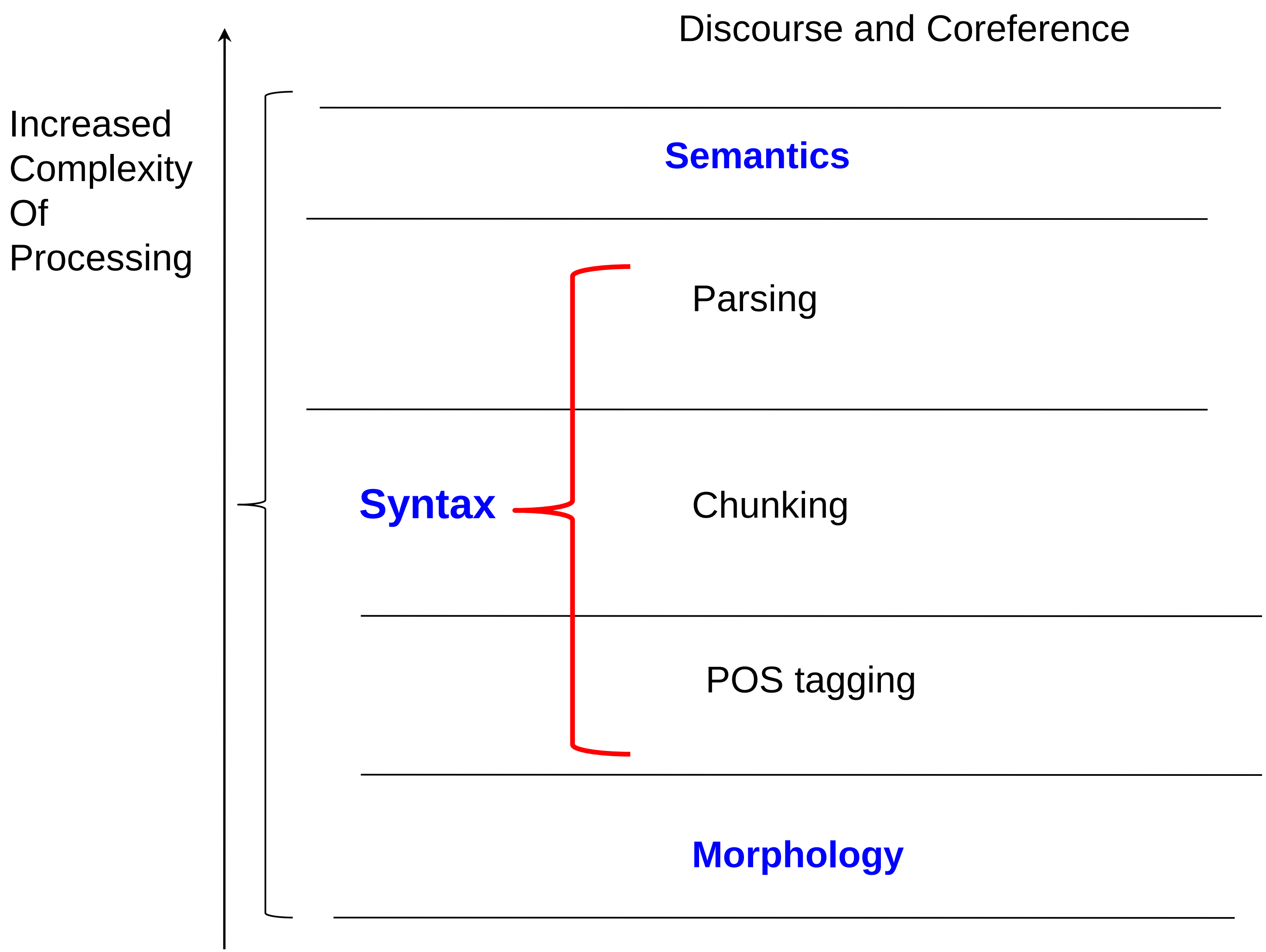

Increased
Complexity
Of
Processing
Discourse and Coreference
Semantics
Parsing
Syntax
Chunking
POS tagging
Morphology

Illustration of Viterbi Decoding for POS tagging

From upcoming book:

Pushpak Bhattacharyya and Aditya Joshi, *Natural Language Processing,* Wiley Eastern, 2022.

Sentence: "*People Dance*"

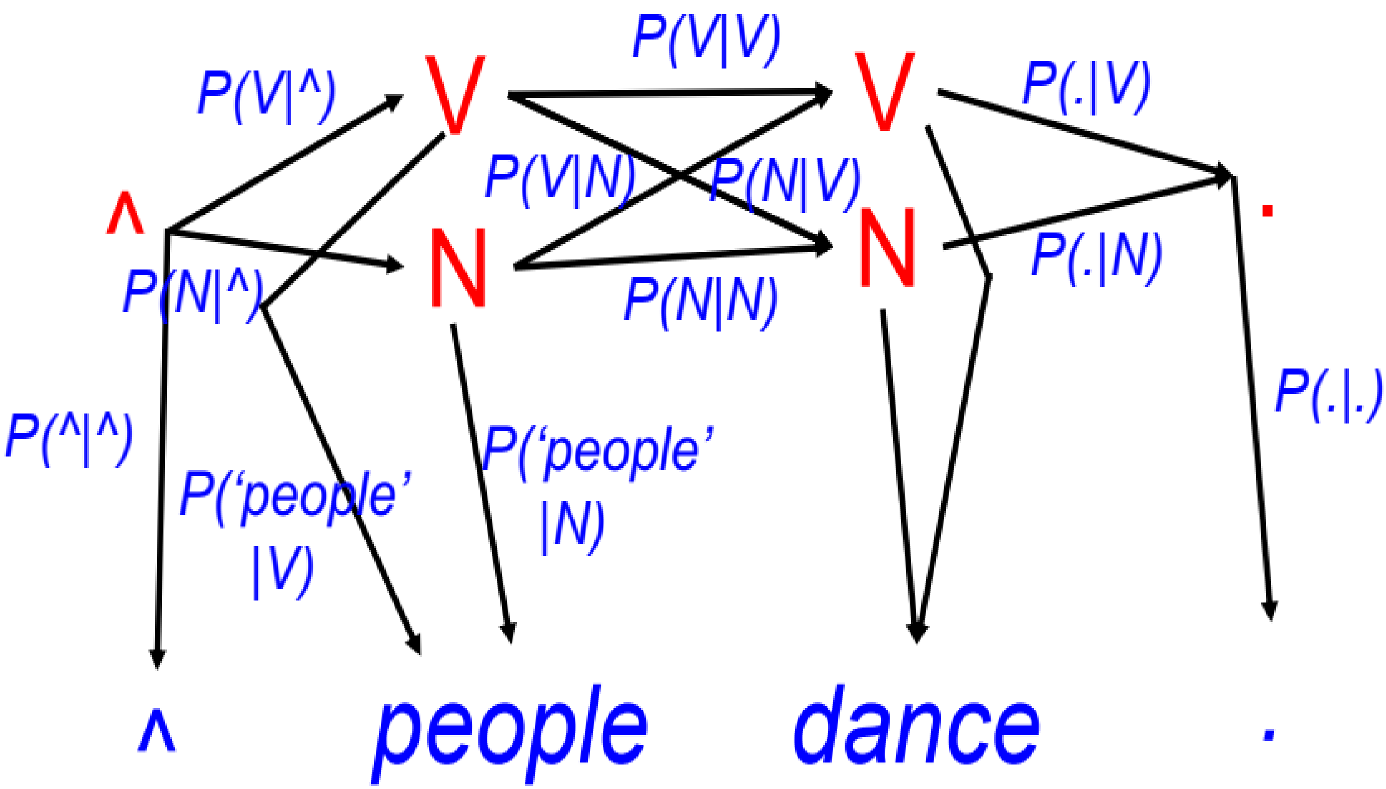
- '*people*' and '*dance*' can both be both nouns and verbs, as in
  - "*old_JJ people_NNS*" ('*people*' as noun)
  - "*township_NN peopled_VBN with soldiers_NNS*" ('*people*' as verb)
- as well as
  - "*rules_NNS of_IN classical_JJ dance_NN*" ('*dance*' as noun)
  - "*will*_VAUX *dance_VB well_RB*" ('*dance*' as verb)

Possible Tags: "^ *people dance .*"

- for simplicity we take single letter tags- *N: noun, V: verb*:
  - ^ *N N .*
  - ^ *N V .*
  - ^ *V N .*
  - ^ *V V .*
- We know that out of these, the second option ^ *N V.* is the correct one. How do we get this sequence?

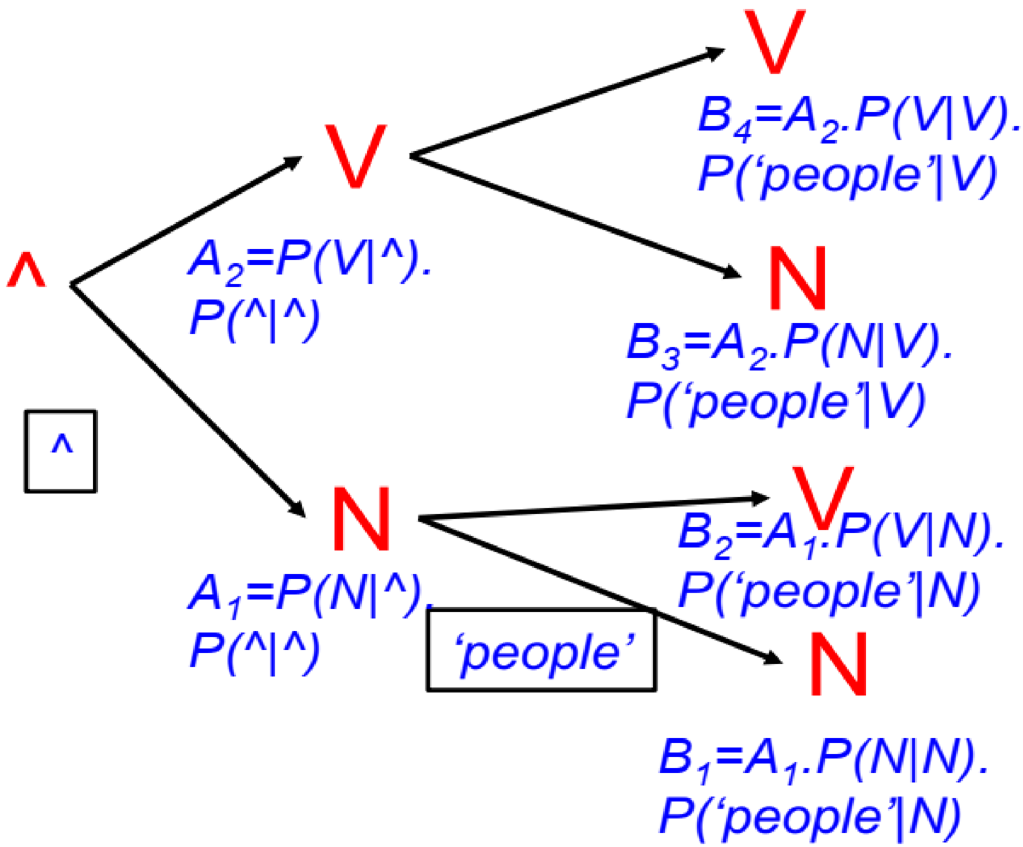
Step-1: Trellis

Columns of tags on each input word with transition arcs going from tags (states) to tags in consecutive columns and output arcs going from tags to words (observations)



Aim: select the highest probability path

From 4 possibilities; *A*s and *B*s are accumulated probabilities



Some numerical values: hypothetical but not unrealistic

- Calculations:
- When it comes to the start of the sentence, most sentences start with a noun. So lets have

  $P(N|\wedge)=0.8$, $P(V|\wedge)=0.2$ and of course $P(\text{'}\wedge\text{'}|\wedge)=1.0$
- Then

$$A_1=0.8,\ A_2=0.2$$

Encounter "people": more probabilities (1/2)

- Transition from *N* to *N* is less common than to *V*.
- Transition from *V* to *V*- as in auxiliary verb to main verb- is quite common (e.g., *is going*).
- *V* to *N* too is common- as in case of a nominal object following the verb *(going home).*
- Following plausible transition probabilities:
  - $P(N|N)=0.2$, $P(V|N)=0.8$, $P(V|V)=0.4$, $P(N|V)=0.6$
- We also need lexical probabilities. 'people' appearing as verb is much less common than its appearing as noun. So let us have

Encounter “people”: more probabilities (2/2)

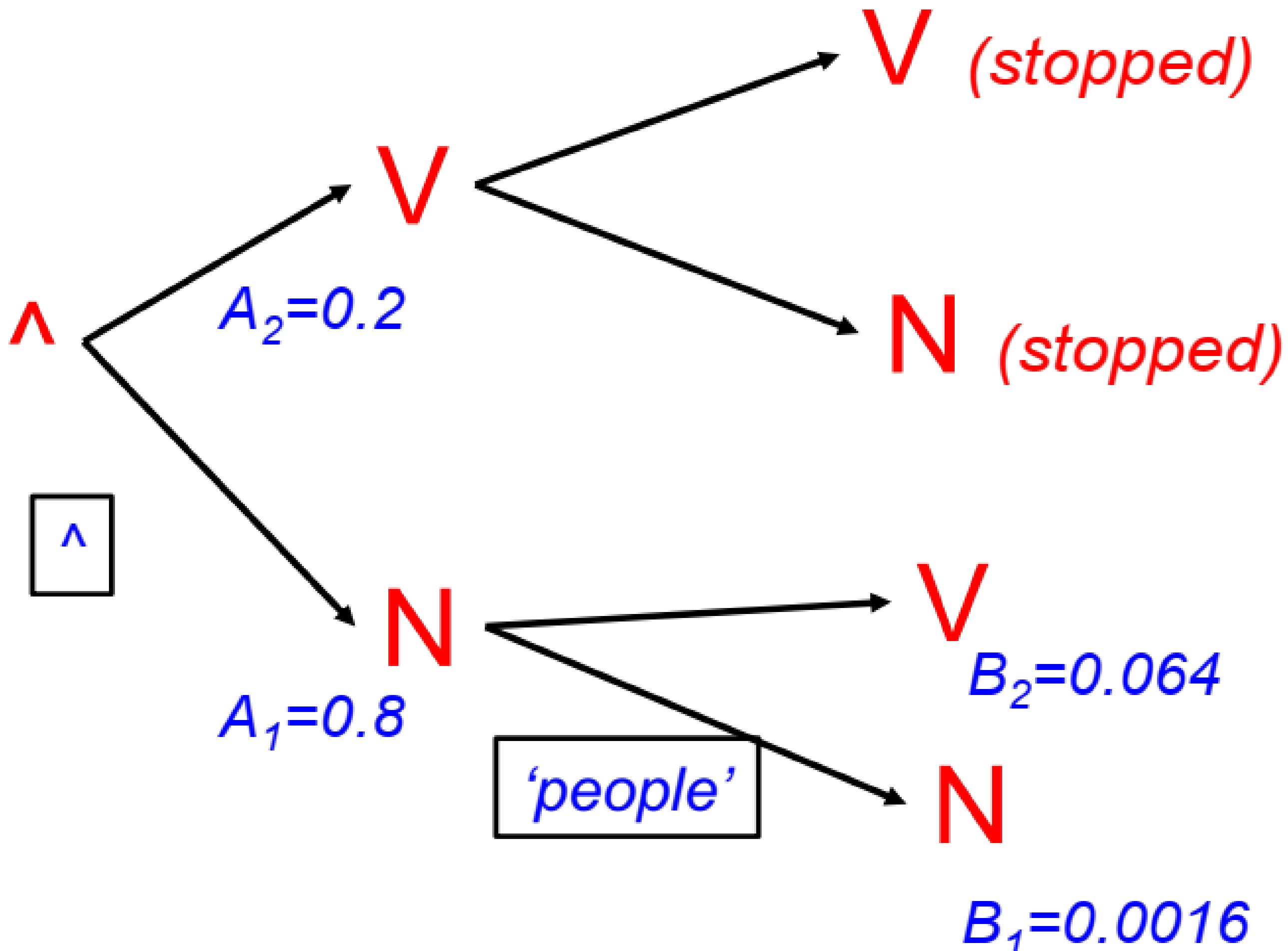
- We also need lexical probabilities. ‘people’ appearing as verb is much less common than its appearing as noun. So let us have
  - *P(‘people’|N)=0.01, P(‘people’|V)=0.001*
- Note: *N N*: golf club, cricket bat, town people- ambiguity “The town people visited was deserted”/”town people will not be able to live here”
- V V combination: Hindi- *has padaa* (laughed suddenly), Bengali- *chole gelo* (went away)

Calculate *B*s

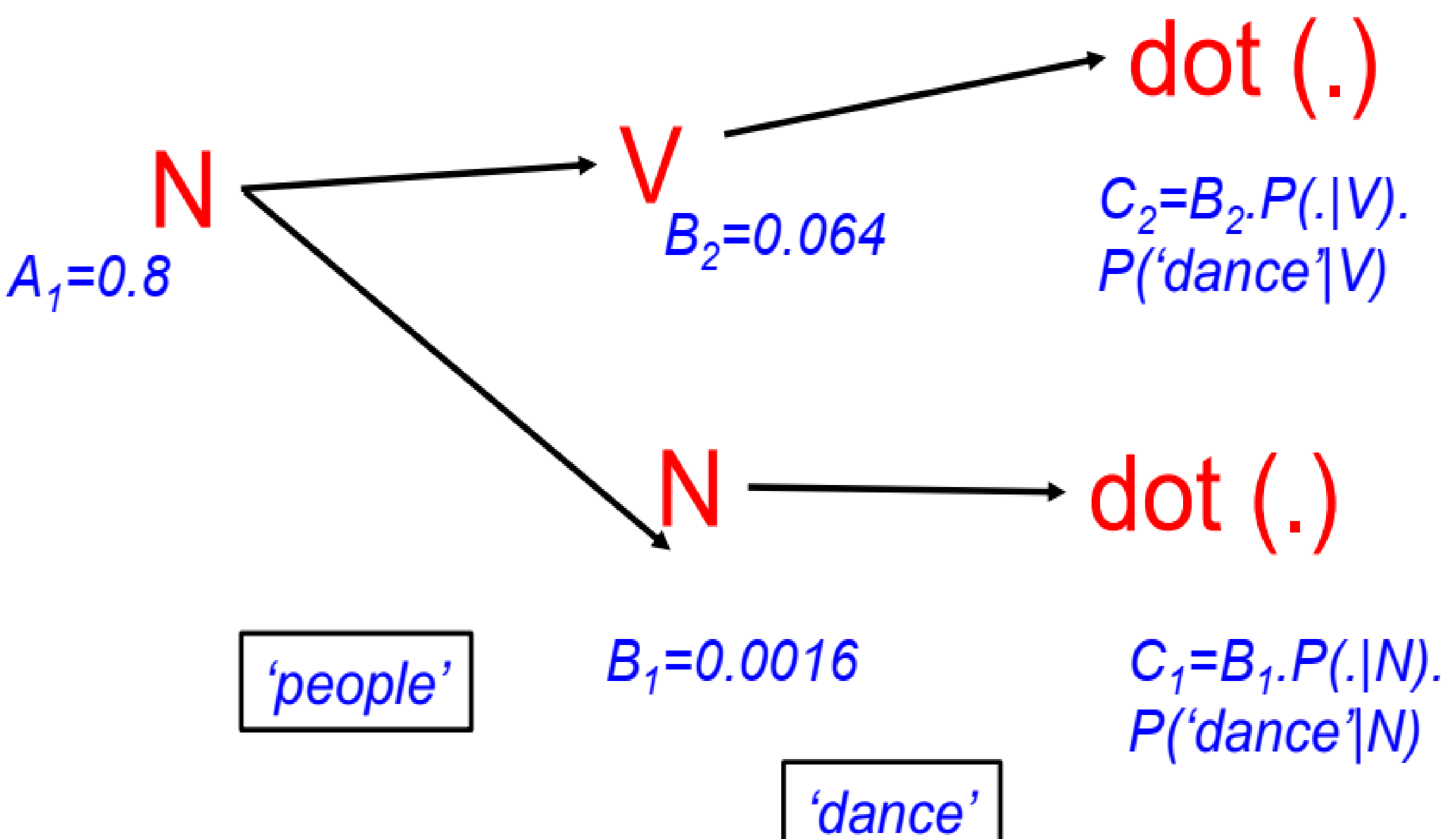
- B_1=0.8.0.2.0.01=0.0016 (approx.)
- B_2=0.8.0.8.0.01=0.064 (approx.)
- B_3=0.2.0.6.0.001=0.00012
- B_4=0.2.0.4.0.001=0.00008

Reduced Viterbi Configuration

- Heart of Decoding → linear time



Next word: ‘*dance*’



More probabilities needed

- We can give equal probabilities to sentences ending in noun and verb. Also, ‘dance’ as verb is more common than noun.

$$P(.|N)=0.5=P(.|V)$$

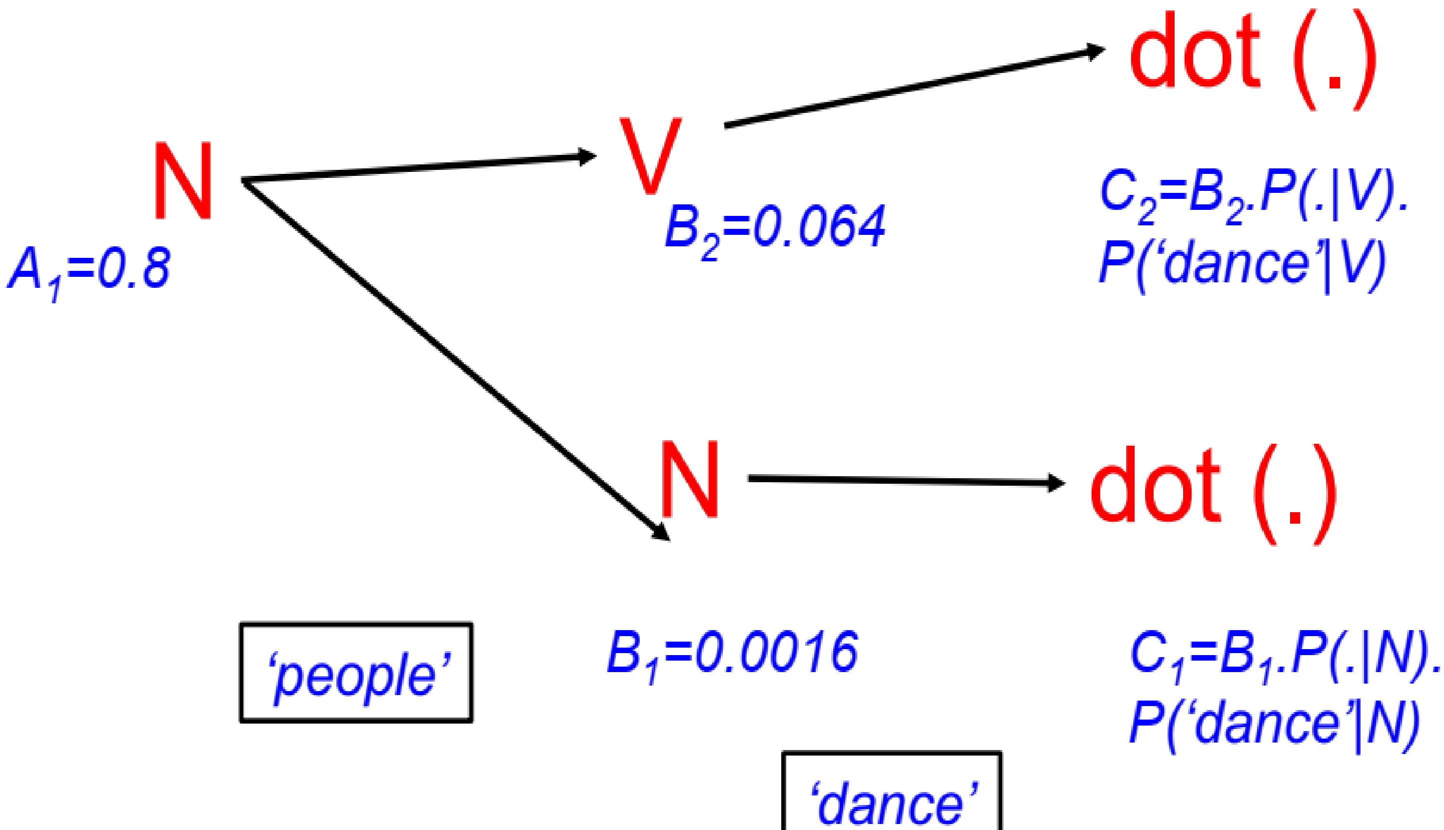
$$P(\text{‘dance’}|N)=0.001$$

$$P(\text{‘dance’}|V)=0.01$$

Best Path: ^ N V .

$$C_1 = 0.0016.0.5.0.001 = 0.0000008$$

$$C_2 = 0.064.0.5.0.01 = 0.00032$$



What does POS tagging Facilitate

Facilitates Chunking: small phrases called **Chunks**

- given the sentence

  The brown fox sat in front of the fence

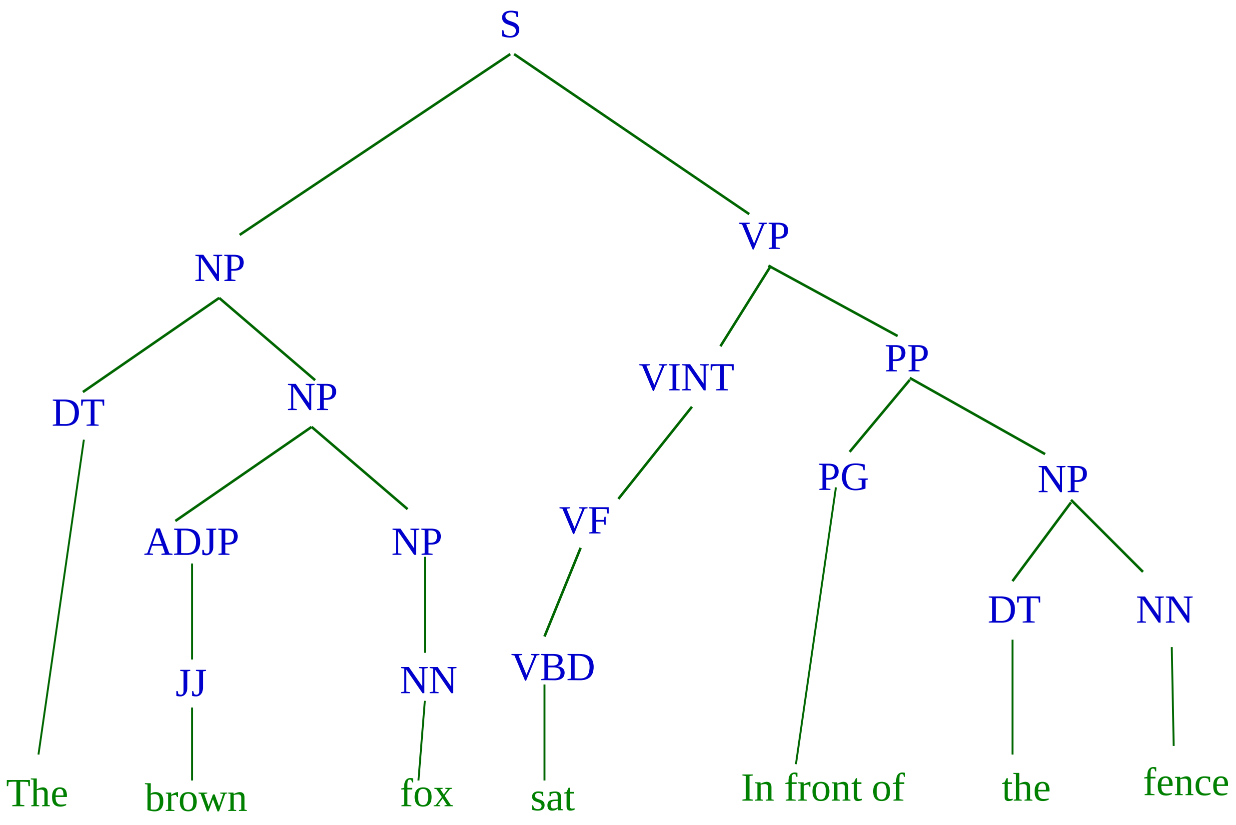
- POS tagged sequence as

  The_DT brown_JJ fox_NN sat_VBD in_IN front_NN of_IN the_DT fence_NN

Chunked sequence as

*The_DT_*B_{NC} *brown_JJ_*I_{NC} *fox_NN_*I_{NC}
*sat_VBD_*B_{VC} *in_IN_*B_{PC} *front_NN_*I_{PC}
*of_IN_*I_{PC} *the_DT_*B_{NC} *fence_NN_*I_{NC}

Deep Parse Tree of *the brown fox sat in front of the fence*



Grammar rules

- S → NP VP
- NP → DT NP | ADJP NP | PP NP | NNS | NN
- ADJP → ADJP JJ | JJ
- PP → PG NP | P NP
- PG → ‘in front of’ | ‘in lieu of’ | ‘with respect to’ | …
- P → ‘in’ | ‘with’ | ‘by’ | …
- NN → ‘fox’ | ‘fence’ | …
- JJ → ‘brown’ | …
- DT → ‘a’ | ‘an’ | ‘the’ | …

- VP → VT NP | VINT PP
- VT → VXG VF | VF
- VINT → VXG VF | VF
- VXG → VXG VX | VX
- VF → VB | VBD | …
- VX → ‘am’ | ‘is’ | ‘shall’ | …
- VB → ‘go’ | ‘see’ | …
- VBD → ‘sat’ | ‘went’ | …
- NN → ‘fox’ | ‘fence’ | …

Discriminative Labelling

Motivation

- HMM based POS tagging cannot handle “free word order” and “agglutination” well
- If *adjective after noun* is equally likely as *adjective before noun*, the transition probability is no better than uniform probability which has high entropy and is uninformative.
- When the words are long strings of many morphemes, POS tagging w/o morph features is highly inaccuarte.

Modelling

$$\prod_{i=0}^{n+1} [P(t_i \mid F_i)]$$

Feature Engineering

- *A. Word-based features*

  f_{21} – dictionary index of the current word ('foxes'): integer

  f_{22} – -do- of the previous word ('brown'): integer

  f_{23} – -do- of the next word ('jumped'): integer

- *B. Part of Speech (POS) tag-based feature*

  f_{24} – index of POS of previous word (here JJ): integer

Feature engineering cntd.

- *C. Morphology-based features*

  - f_{25}– does the current word (‘foxes’) have a noun suffix, like ‘s’, ‘es’, ‘ies’, etc.: 1/0- here the value is
  - f_{26}– does the current word (‘foxes’) have a verbal suffix, like ‘d’, ‘ed’, ‘t’, etc.: 1/0- 0
  - f_{27} and f_{28} for ‘brown’ like for ‘foxes
  - f_{29} and $f_{2,10}$ for ‘jumped’ like for ‘foxes; here $f_{2,10}$ is 1 (jumped has ‘ed’ as suffix)

A note of morph features (1/2)

- Morphology features can be fairly open ended, large in number and complex depending on the language under consideration.
- Dravidian languages, Tibeto-Burman languages, Arabic, Hungarian, Turkish, Finnish and so on are morphologically complex.

A note of morph features (1/2)

- Used with dexterity, they can disambiguate POS tags with very high degree of certainty.

- For example, the '*unnu*' suffix in the Malayalam word '*ceyy-unnu*': English- '*does, is doing*' is a sure-shot identifier of verb POS (VBS).