CS60075
Natural Language Processing
Autumn 2020

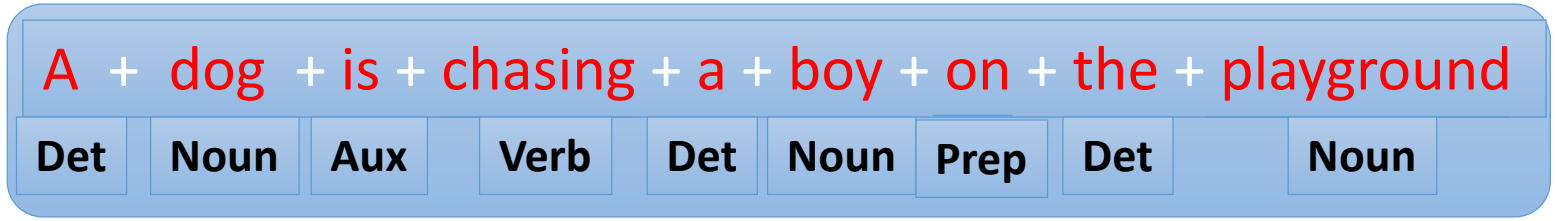
Module 4: Part 2
Introduction to POS Tagging

Sep 24 2020

Part-of-Speech tagging

Input: A sequence of tokens (e.g., a sentence, tweet, search queries).
Output: An assignment of POS tags for each word in the input
the most likely tag sequence

A	tall	boy	ran	home	.
DT	JJ	NN	VBD	NN	.



Part Of Speech Tagging

Annotate each word in a sentence with a part-of-speech marker with the most likely tag sequence

John	saw	the	saw	and	decided	to	take	it	to	the	table.
NNP	VBD	DT	NN	CC	VBD	TO	VB	PRP	IN	DT	NN

A	tall	boy	ran	home	.
DT	JJ	NN	VBD	NN	.

Why do POS tagging?

- Text-to-speech
- Can write regular expressions over POS tags to identify phrases etc.

  e.g., (Det) Adj* N+ over the output for noun phrases, etc.
- Assigning classes to words or phrases is quite useful
  - Noun, Verb, Adjective, …
  - Subject, Verb, Direct Object, Indirect Object, …
  - Person, Organization, Date, …
  - Drug, Disease, Side-effect, …

Current state

- State-of-the-art techniques achieve > 98% accuracy on newswire texts.
- POS tagging in resource poor languages is harder (~87%)
- POS tagging for Tweets is harder (state-of-the art ~ 88 %)

Named entity recognition

- Determine text mapping to proper names
  - Task: what is the most likely mapping

Its initial Board of Visitors included U.S. Presidents Thomas Jefferson, James Madison, and James Monroe.

Its initial Board of Visitors included U.S. Presidents Thomas Jefferson, James Madison, and James Monroe.

Organization, Location, Person

Parts of speech

Parts of speech are constructed by grouping words that function similarly:

- with respect to the words that can occur nearby
- and by their morphological properties

The man _____ all the way home.

- Aristotle (384–322 BCE): the idea of having parts of speech a.k.a lexical categories, word classes, “tags”, POS
- Dionysius Thrax of Alexandria (c. 100 BCE) : 8 parts of speech
  - noun, verb, article, adverb, preposition, conjunction, participle, pronoun

English parts of speech

- 8 parts of speech?
  - Noun (person, place or thing)
  - Verb (actions and processes)
  - Adjective (modify nouns)
  - Adverb (modify verbs)
  - Preposition (on, in, by, to, with)
  - Determiners (a, an, the, what, which, that)
  - Conjunctions (and, but, or)
  - Particle (off, up)

Brown corpus: 87 POS tags

Penn Treebank: ~45 POS tags

English Parts of Speech

Noun: Syntactic Function: Subjects or Objects of verbs.
Semantic Type: Person, place or thing.
Sub-categories:
Singular (NN): dog, fork
Plural (NNS): dogs, forks
Proper (NNP, NNPS): John, Springfields
Personal pronoun (PRP): I, you, he, she, it
Wh-pronoun (WP): who, what

Verb: Syntactic Function: predicate, heads a verb phrase.
Semantic Type: actions or processes.
Sub-categories:
Base, infinitive (VB): eat
Past tense (VBD): ate
Gerund (VBG): eating
Past participle (VBN): eaten
Non 3rd person singular present tense (VBP): eat
3rd person singular present tense: (VBZ): eats
Modal (MD): should, can
To (TO): to (to eat)

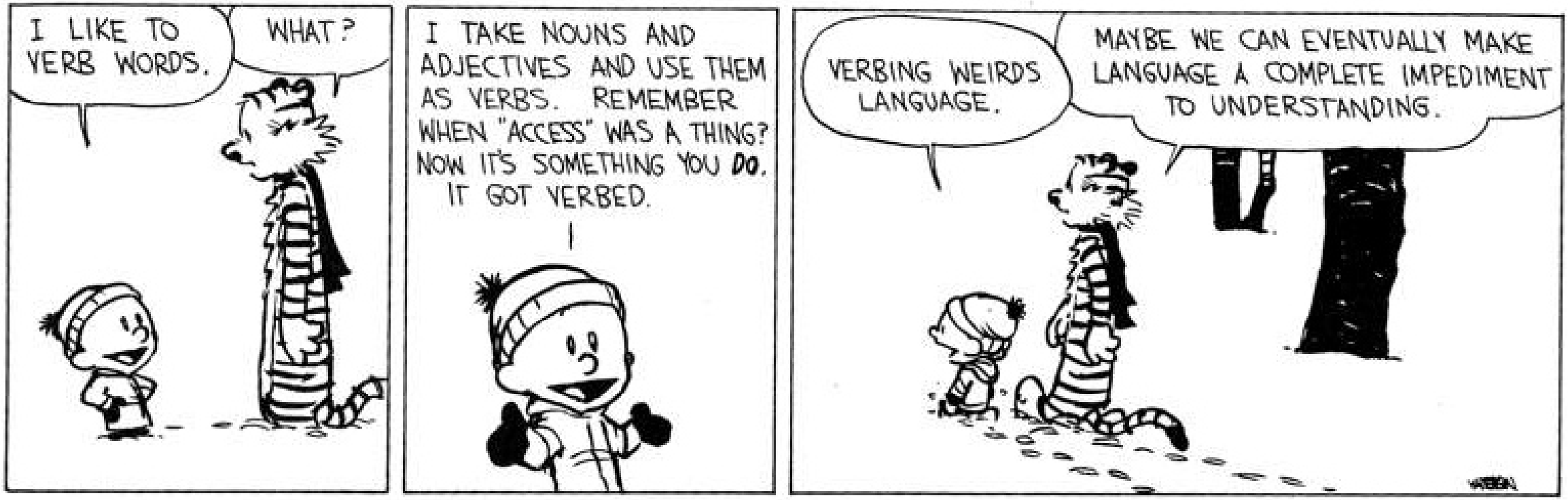
English Parts of Speech

- Adjective (modify nouns)
  - Basic (JJ): red, tall
  - Comparative (JJR): redder, taller
  - Superlative (JJS): reddest, tallest
- Adverb (modify verbs)
  - Basic (RB): quickly
  - Comparative (RBR): quicker
  - Superlative (RBS): quickest
- Preposition (IN): on, in, by, to, with
- Determiner:
  - Basic (DT) a, an, the
  - WH-determiner (WDT): which, that
- Coordinating Conjunction (CC): and, but, or,
- Particle (RP): off (took off), up (put up)

Closed vs. Open Class

- ***Closed class*** categories are composed of a small, fixed set of grammatical function words for a given language.
  - **Pronouns, Prepositions, Modals, Determiners, Particles, Conjunctions**

- ***Open class*** categories have large number of words and new ones are easily invented.
  - Nouns: Googler, textlish
  - Verbs: You can verb many nouns. E.g., Google it.
  - Adjectives: Many nouns can be made into adjectives. E.g., geeky
  - Abverb: e.g., Webly supervised learning


I LIKE TO VERB WORDS.
WHAT?
I TAKE NOUNS AND ADJECTIVES AND USE THEM AS VERBS. REMEMBER WHEN "ACCESS" WAS A THING? NOW IT'S SOMETHING YOU DO. IT GOT VERBED.
VERBING WEIRDS LANGUAGE.
MAYBE WE CAN EVENTUALLY MAKE LANGUAGE A COMPLETE IMPEDIMENT TO UNDERSTANDING.

Part Of Speech Tagging

- Annotate each word in a sentence with a part-of-speech marker.

John	saw	the	saw	and	decided	to	take	it	to	the	table.
NNP	VBD	DT	NN	CC	VBD	TO	VB	PRP	IN	DT	NN

UDEP POS tags

Open class words	Closed class words	Other
ADJ	ADP	PUNCT
ADV	AUX	SYM
INTJ	CCONJ	X
NOUN	DET	
PROPN	NUM	
VERB	PART	
	PRON	
	SCONJ	

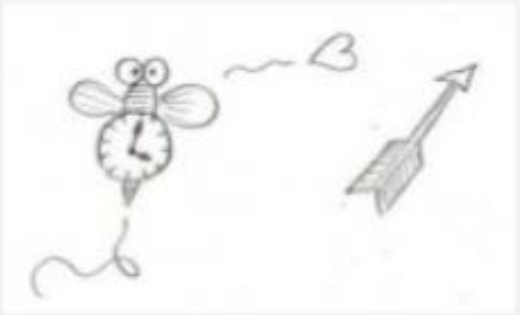
Ambiguity in POS Tagging

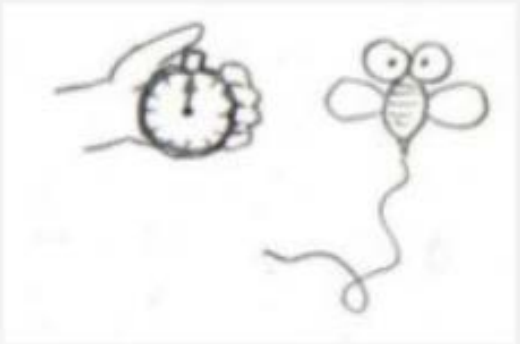
Like most language components, the challenge with POS tagging is ambiguity

- "Like" can be a verb or a preposition
  - I like/VBP candy.
  - Time flies like/IN an arrow.

- What is the POS for "back"?
  - The *back* door
  - On my *back*
  - Win the voters *back*
  - Promised to *back* the bill

Time flies like an arrow





Time flies like an arrow.

Time flies like an arrow.

Time flies like an arrow.

Ambiguity in POS tagging

Brown corpus analysis

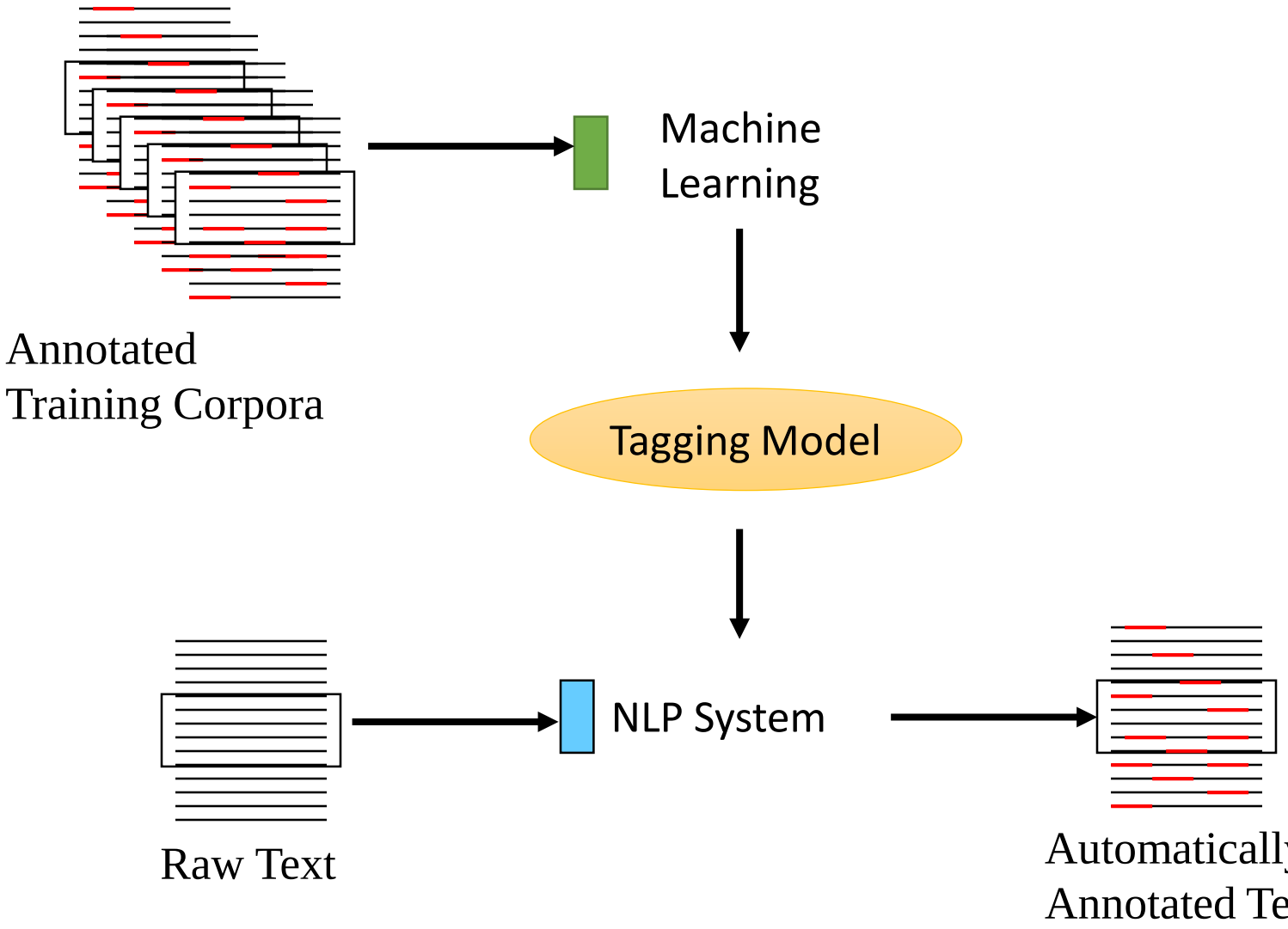
- 11.5% of word types are ambiguous, but…
- 40% of word appearances are ambiguous
- Unfortunately, the ambiguous words tend to be the more frequently used words

- Cannot specify POS tags for all words
  - New words appear all the time.

POS Tagging Approaches

- **Rule-Based**: Human crafted rules based on lexical and other linguistic knowledge.
- **Learning-Based**: Trained on human annotated corpora like the Penn Treebank.
  - **Statistical models**: Hidden Markov Model (HMM), Maximum Entropy Markov Model (MEMM), Conditional Random Field (CRF)
  - **Rule learning**: Transformation Based Learning (TBL)
  - **Neural networks**: Recurrent networks like Long Short Term Memory (LSTMs)

Learning Approach



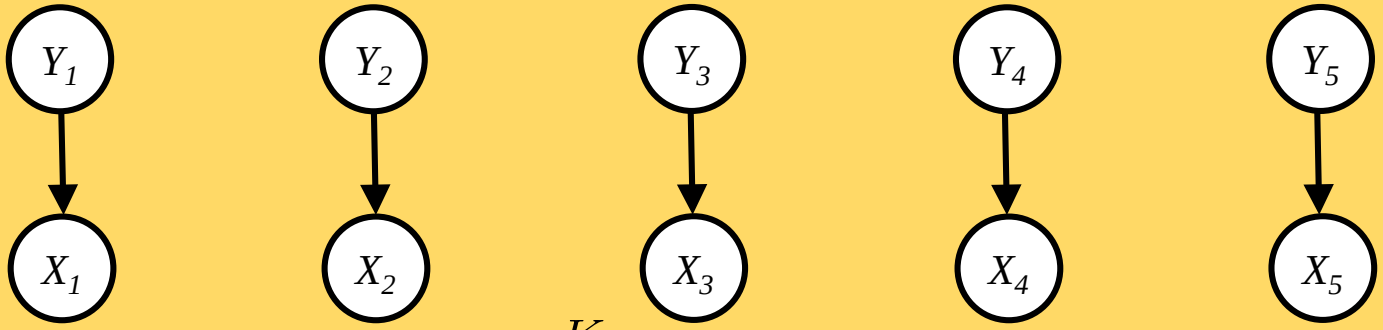
Problem Formulations

1. Baseline
   - Assign most frequent tag for each word based on training.
2. Hidden Markov Models (HMM)
   - Previous tag and current word should influence current tag
3. Feature rich model
   - Maximum Entropy (Maxent) model
4. Discriminative model
   - Conditional Random Field (CRF)

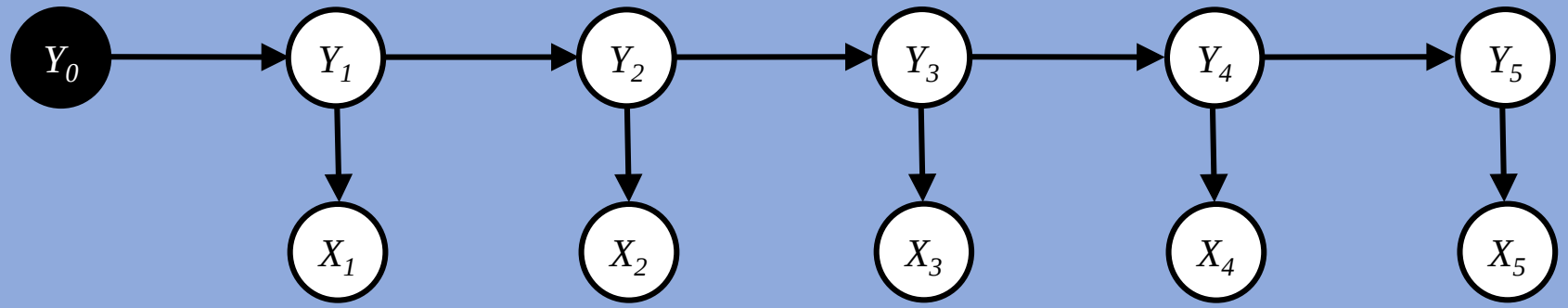
Hidden Markov Model

- Probabilistic generative model for sequences.
- Assume an underlying set of ***hidden*** (unobserved) states in which the model can be (e.g. parts of speech).
- Assume probabilistic transitions between states over time (e.g. transition from POS to another POS as sequence is generated).
- Assume a ***probabilistic*** generation of tokens from states (e.g. words generated for each POS).
- Assume current state is dependent only on the previous state.

From NB to HMM

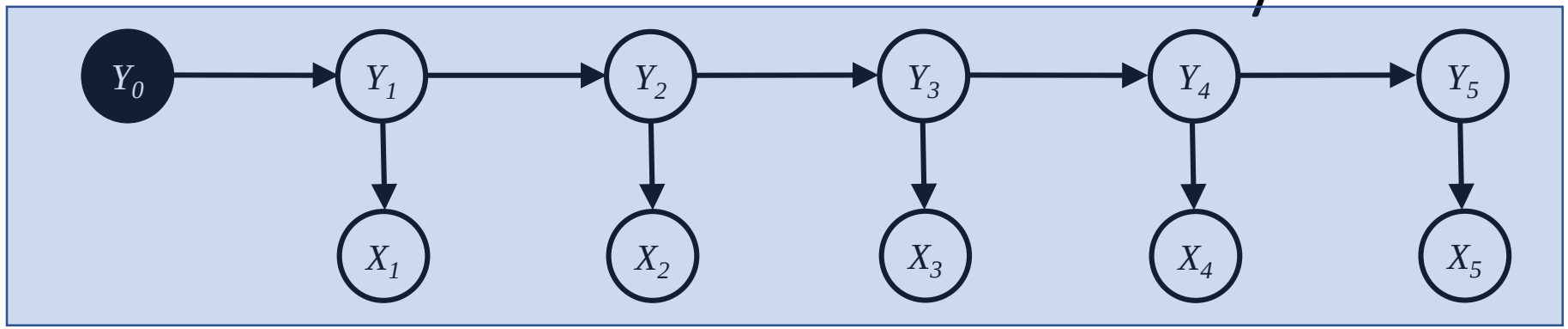


"Naïve Bayes": $$P(\mathbf{X}, \mathbf{Y}) = \prod_{k=1}^{K} P(X_k|Y_k)p(Y_k)$$



HMM: $$P(\mathbf{X}, \mathbf{Y}) = \prod_{k=1}^{K} P(X_k|Y_k)p(Y_k|Y_{k-1})$$

POS HMMs: A Generative Story



A probabilistic process that generates sentences.

a) Switches through a **finite set of POS states** probabilistically.

e.g., Switches from state i to j with probability $Pr(S_j \mid S_i)$

b) In each state the machine emits some word with a probability that is specific to the state.

e.g. Pr(dog | state 1) = 0.63

Pr(dog | state 2) = 0.21

Formal Definition of an HMM

- A set of *N +2* states $S = \{s_0, s_1, s_2, \ldots s_N, s_F\}$
- A set of *M* possible observations $O = \{o_1, o_2, \ldots, o_M\}$
- A state transition probability distribution $A = \{a_{ij}\}$

$$a_{ij} = P(q_{t+1} = s_j \mid q_t = s_i)$$

$$\sum_{j=1}^{N} a_{ij} + a_{iF} = 1 \quad 0 \le i \le N$$

- Observation probability distribution, $B = \{b_0(k), b_1(k), b_2(k), \ldots, b_F(k)\}$

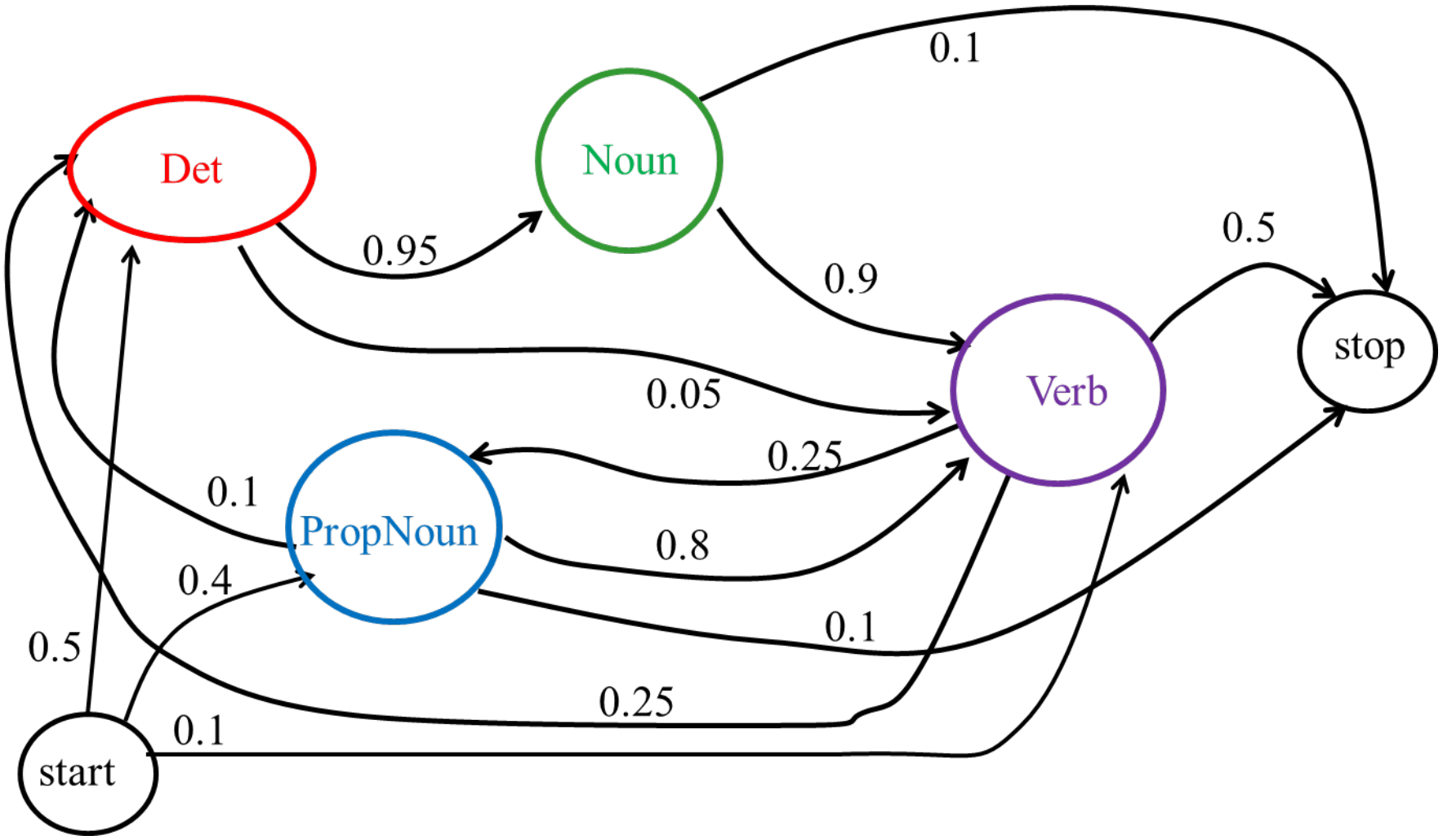
$$b_j(k) = P(v_k \text{ at } t \mid q_t = s_j)$$

- Parameters of the model $\lambda = \{A, B\}$

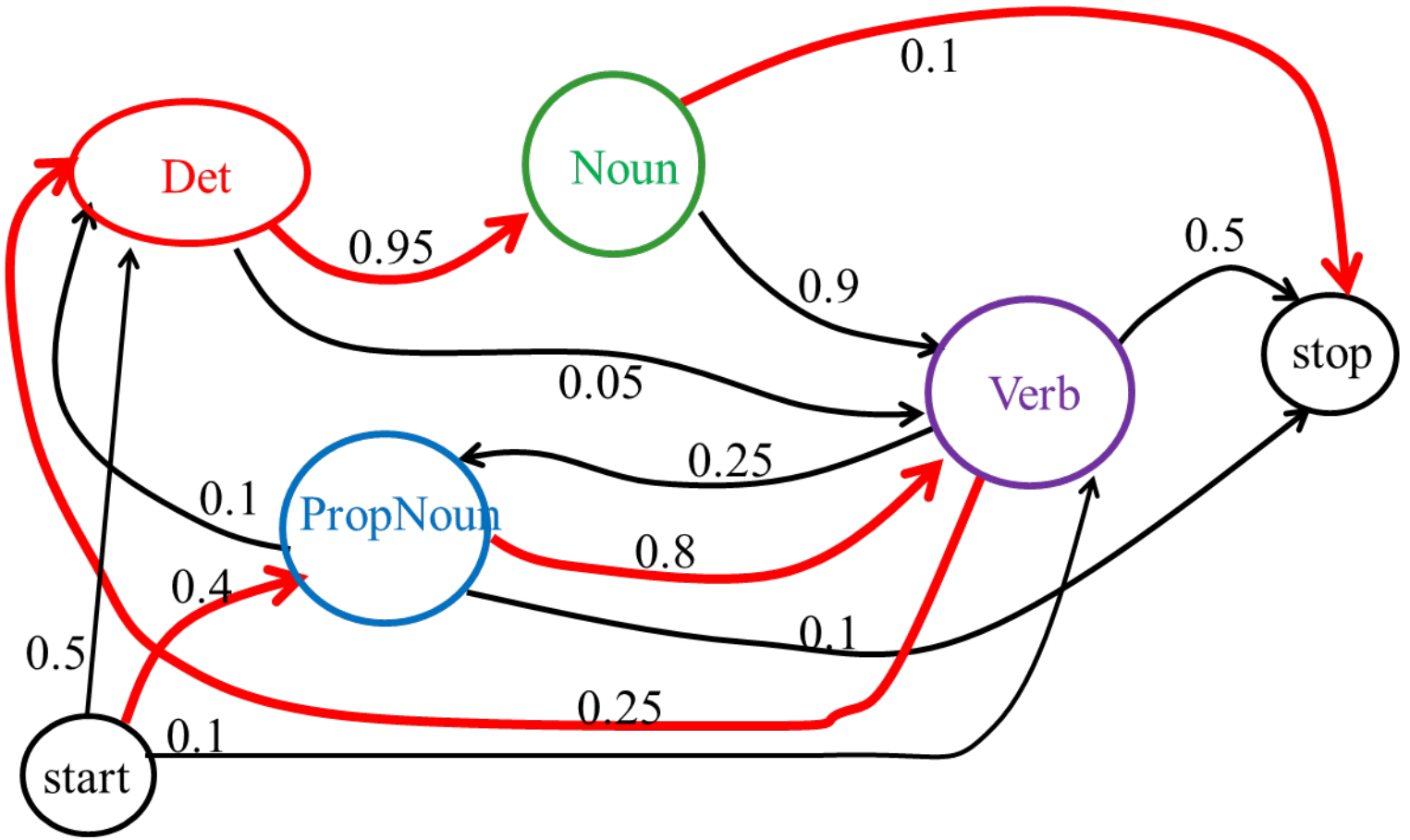
Three Useful HMM Tasks

1. Observation Likelihood (Evaluation): given a model and an output sequence, what is the probability that the model generated that output?
2. Most likely state sequence (Decoding): given a model and an output sequence, what is the most likely state sequence through the model that generated the output?
3. Maximum likelihood training (Learning): given a model and a set of observed sequences, how do we set the model's parameters so that it has a high probability of generating those sequences?

Sample Markov Model for POS

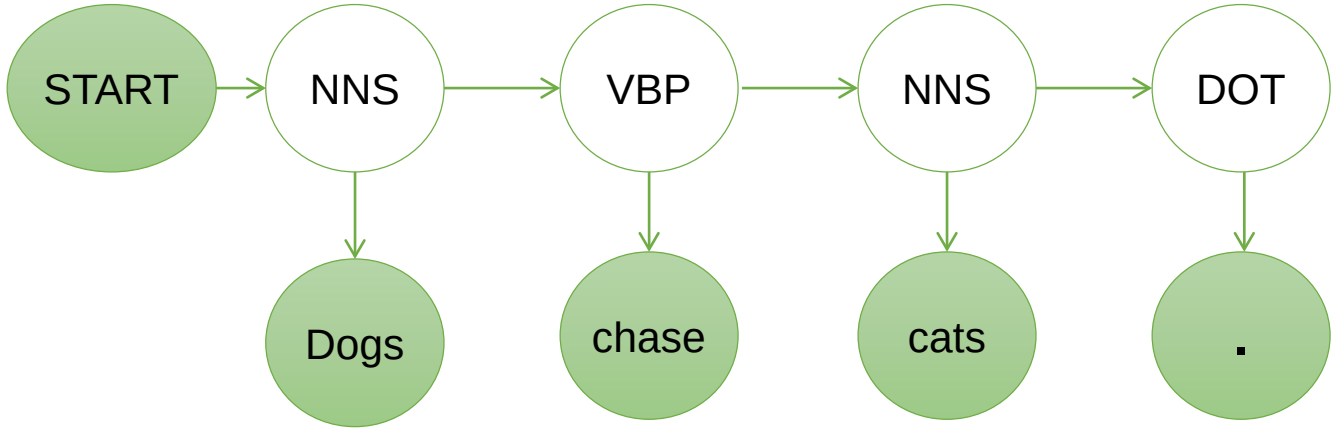


Sample Markov Model for POS



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

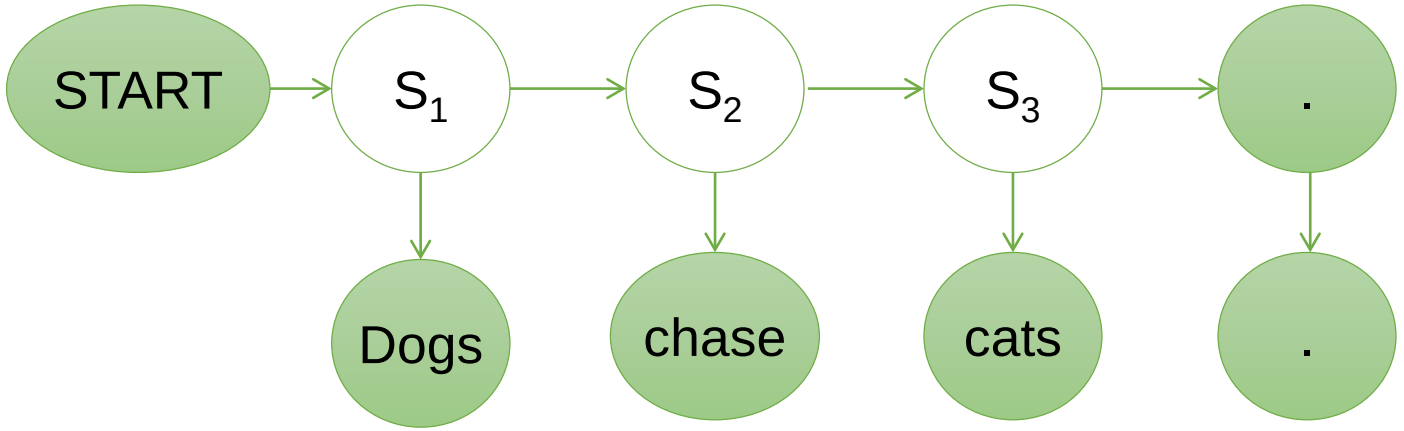
Observation Likelihood



Given the transition and emission tables:
What is the probability of observing the sentence given the model?

Pr(Dogs chase cats.) = Pr(NNS|START) x Pr(Dogs|NNS)
x Pr(VBP|NNS) x Pr(chase|VBP)
x Pr(NNS|VBP) x Pr(cats|NNS)
x Pr(DOT | NNS) x Pr(. | DOT)

Decoding



Given the transition and emission probabilities tables , λ = {A, B}

What are the (most likely) hidden states S1, S2, and S3 that the machine transitioned through in order to produce the observed sentence?

Most Likely State Sequence (Decoding)

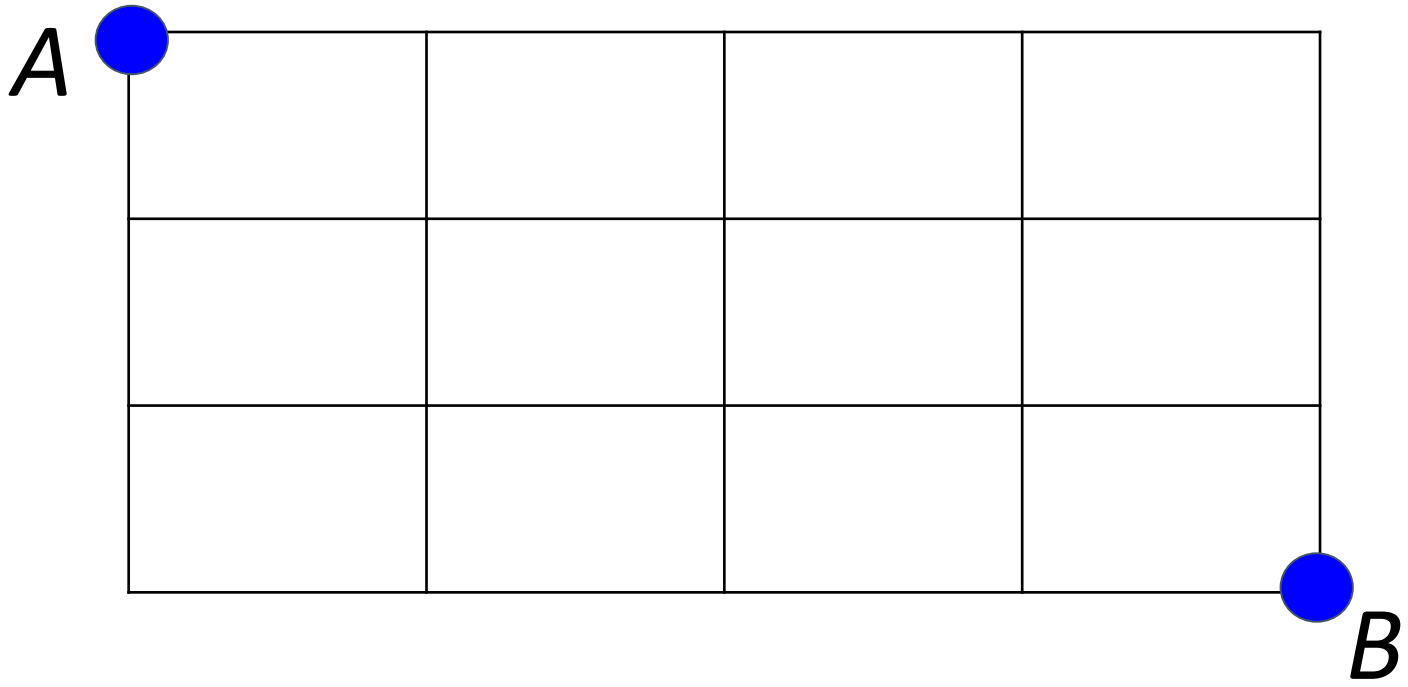
- Used for sequence labeling.
  - Assume each state corresponds to a tag.
  - Determine the globally best assignment of tags to all tokens in a sequence.
  - Uses a principled approach grounded in probability theory.

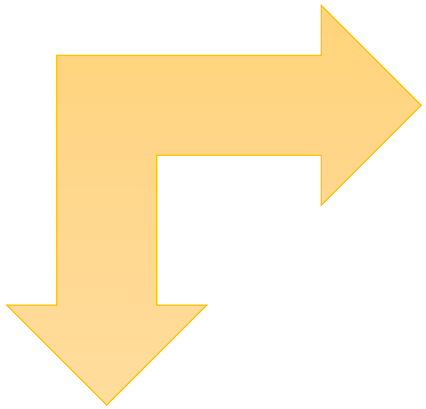
Viterbi Algorithm

- Main idea: using previous calculations to get new results
- Uses a table to store intermediate values
- Approach:
  - Compute the likelihood of the observation sequence
  - By summing over all possible hidden state sequences
  - But doing this efficiently

Viterbi Algorithm: An example

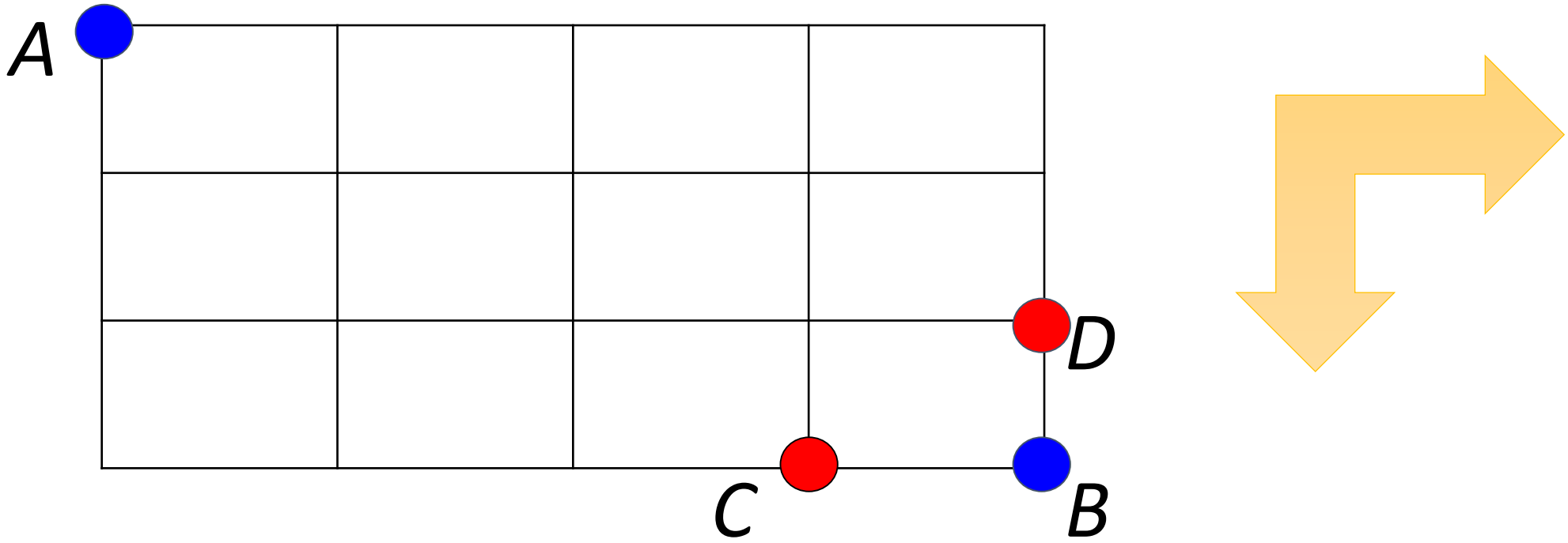
How many distinct ways exists from A to B? With only Right and down movements





Viterbi Algorithm: An example

Wouldn’t be easier to know ways from A to C and D first?



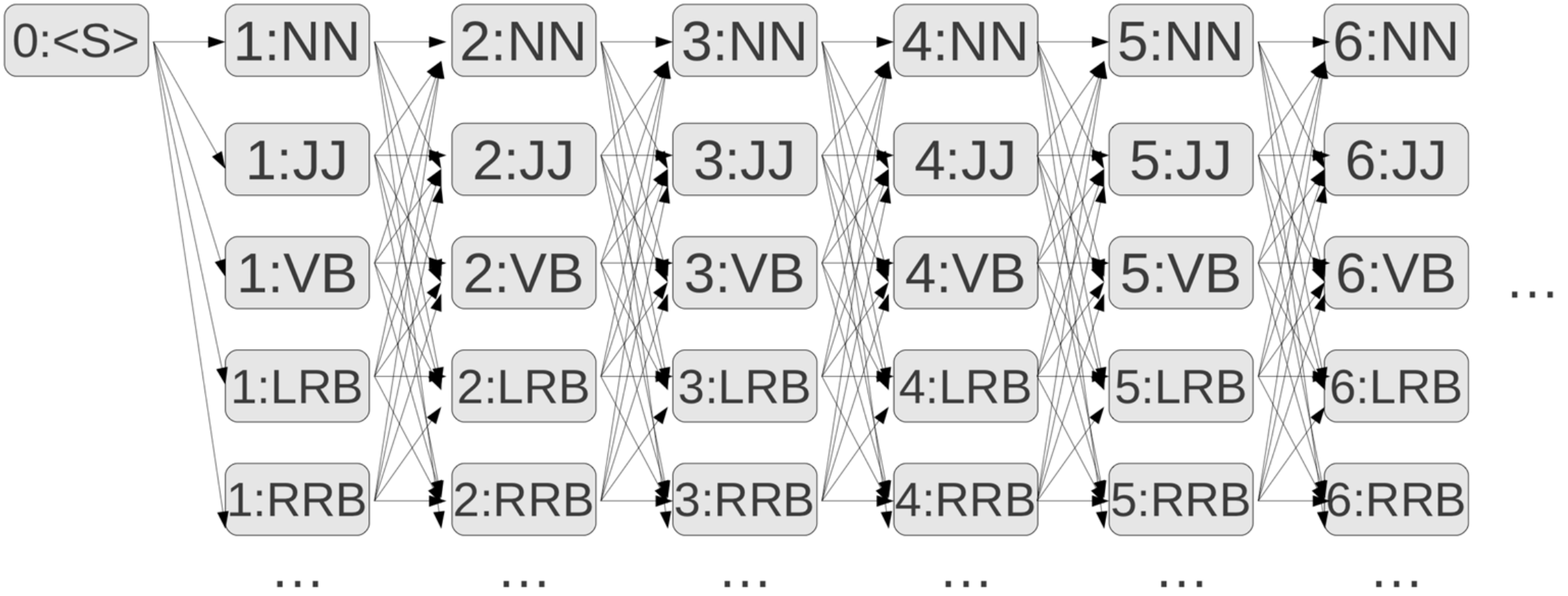
Viterbi Algorithm in POS tagging:

- Dynamical programming algorithm that allows us to compute the most probable path.
- Here,

$$P(x,y) = P(x|y)P(y) = \Pi_i P(x_i|y_i) \bullet \Pi_i P(y_i|y_{i-1})$$

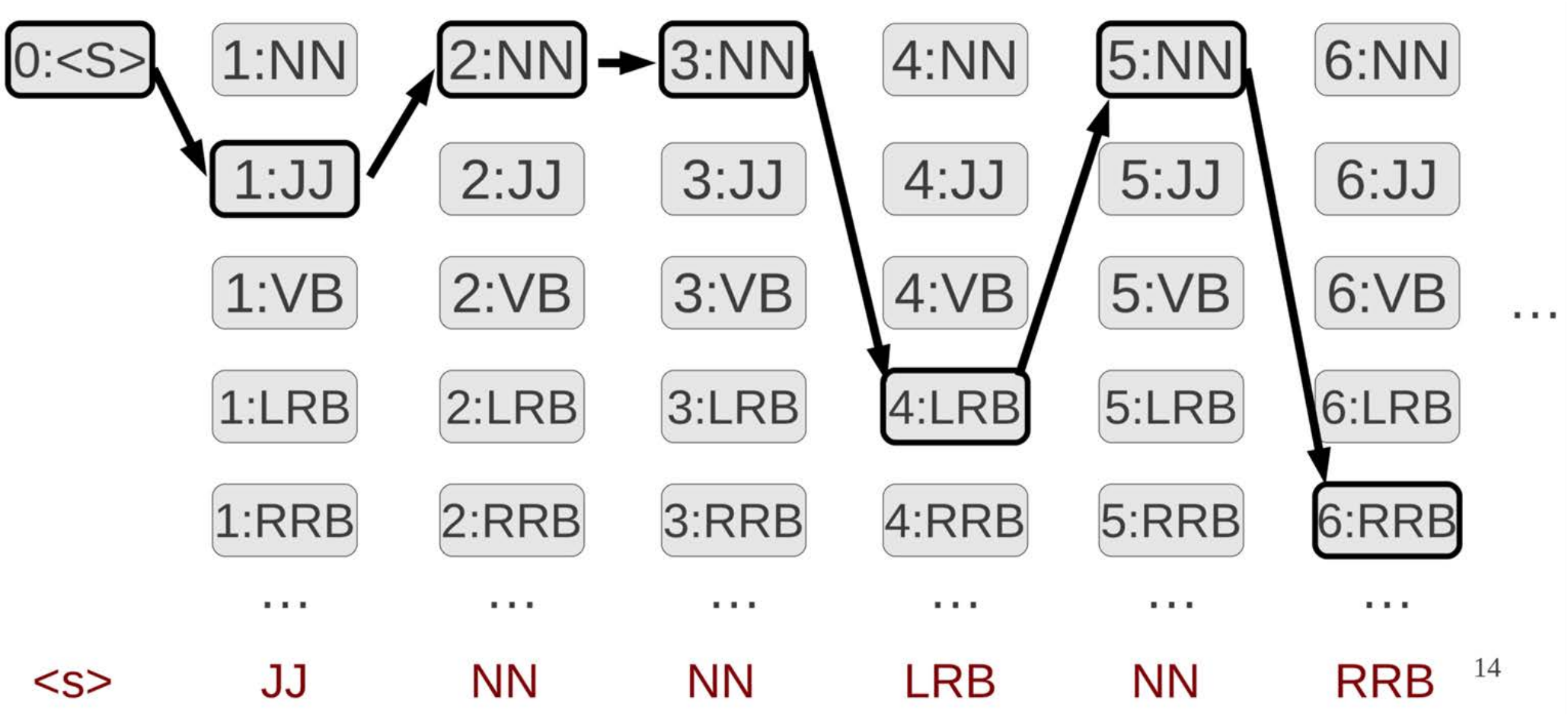
- Therefore a recursive algorithm can be proposed

How To be Recursive?



How To be Recursive?

natural language processing (nlp)



How To be Recursive?

- Assume a sequence of words $o_1, o_2, \dots, o_t$ with corresponding tags $q_1, q_2, \dots, q_t$ (states) .
- Let's define the final state as s_j, i.e., $q_t = s_j$
- We would like to calculate

$$P(O, Q) = P(q_0, q_1, q_2, \dots, q_t = s_j, o_1, o_2, \dots, o_t)$$

- Let's define $v_t(j) = \max_{q_0, q_1, \dots, q_{t-1}} P(O, Q)$
- $v_t(j)$ is the probability of the most probable path accounting for the first t observations and ending in state s_j

Viterbi Scores

- Recursively compute the probability of the most likely subsequence of states that accounts for the first t observations and ends in state s_j.

$$v_t(j) = \max_{q_0, q_1, \ldots, q_{t-1}} P(q_0, q_1, \ldots, q_{t-1}, o_1, \ldots, o_t \; q_t = s_j | \lambda)$$

- Also record "backpointers" that subsequently allow backtracing the most probable state sequence.
  - $bt_t(j)$ stores the state at time $t - 1$ that maximizes the probability that system was in state s_j at time t (given the observed sequence).

Computing the Viterbi Scores

$$v_t(j) = \max_{q_0, q_1, \ldots, q_{t-1}} P(q_0, q_1, \ldots, q_{t-1}, o_1, \ldots, o_t, q_t = s_j | \lambda)$$

- Initialization

$$v_1(j) = a_{0j} b_j(o_1) \quad 1 \leq j \leq N$$

- Recursion

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i) a_{ij} b_j(o_t) \quad 1 \leq j \leq N, \quad 1 < t \leq T$$

- Termination

$$P *= v_{T+1}(s_F) = \max_{i=1}^{N} v_T(i) a_{iF}$$

Computing the Viterbi Backpointers

- Initialization

$$bt_1(j) = s_0 \quad 1 \leq j \leq N$$

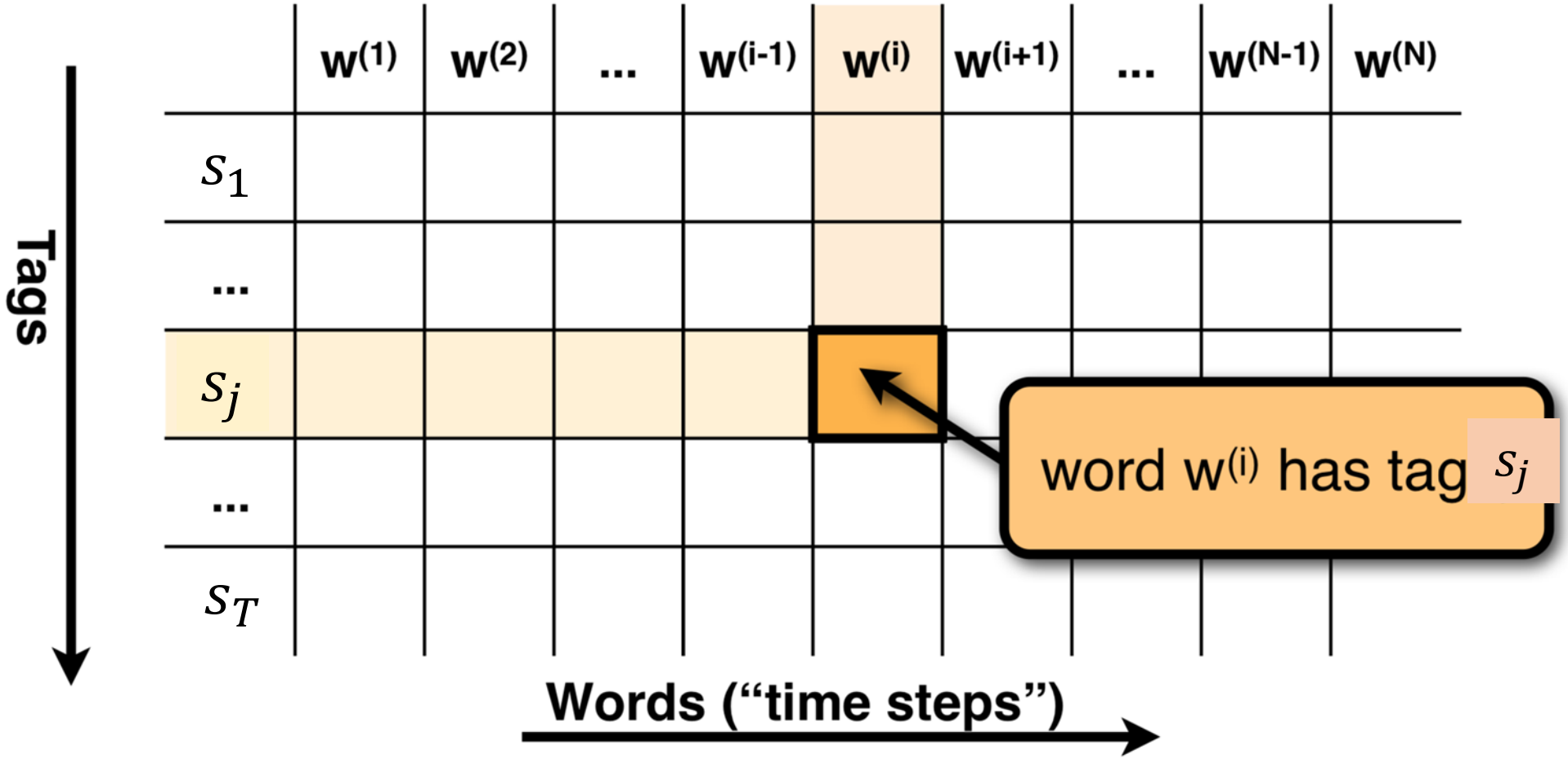
- Recursion

$$bt_t(j) = \underset{i=1}{\overset{N}{\operatorname{argmax}}} v_{t-1}(i) a_{ij} b_j(o_t) \quad 1 \leq j \leq N, \ 1 \leq t \leq T$$

- Termination

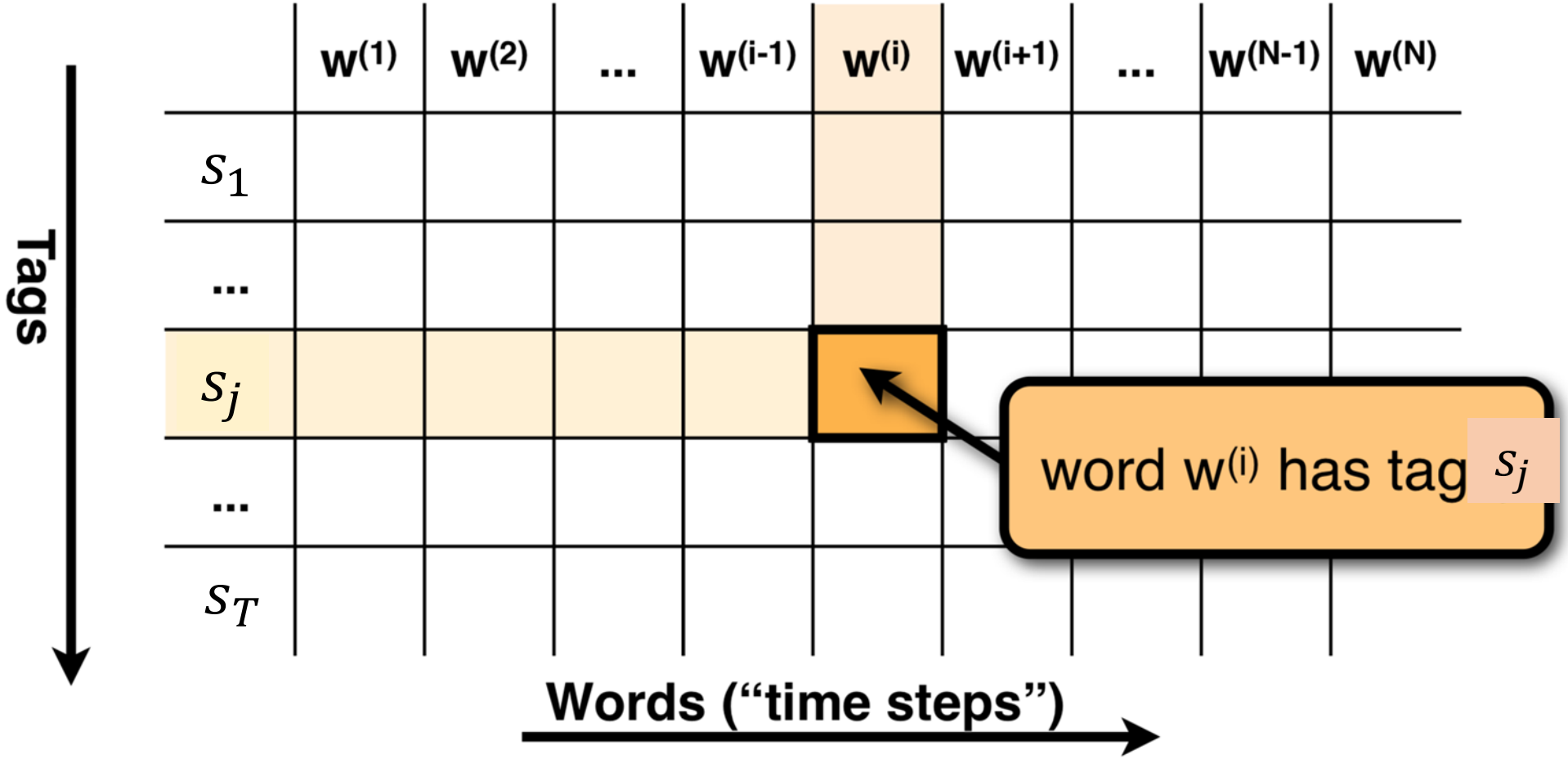
$$q_T *= bt_{T+1}(s_F) = \underset{i=1}{\overset{N}{\operatorname{argmax}}} v_T(i) a_{iF}$$

Final state in the most probable state sequence. Follow backpointers to initial state to construct full sequence.

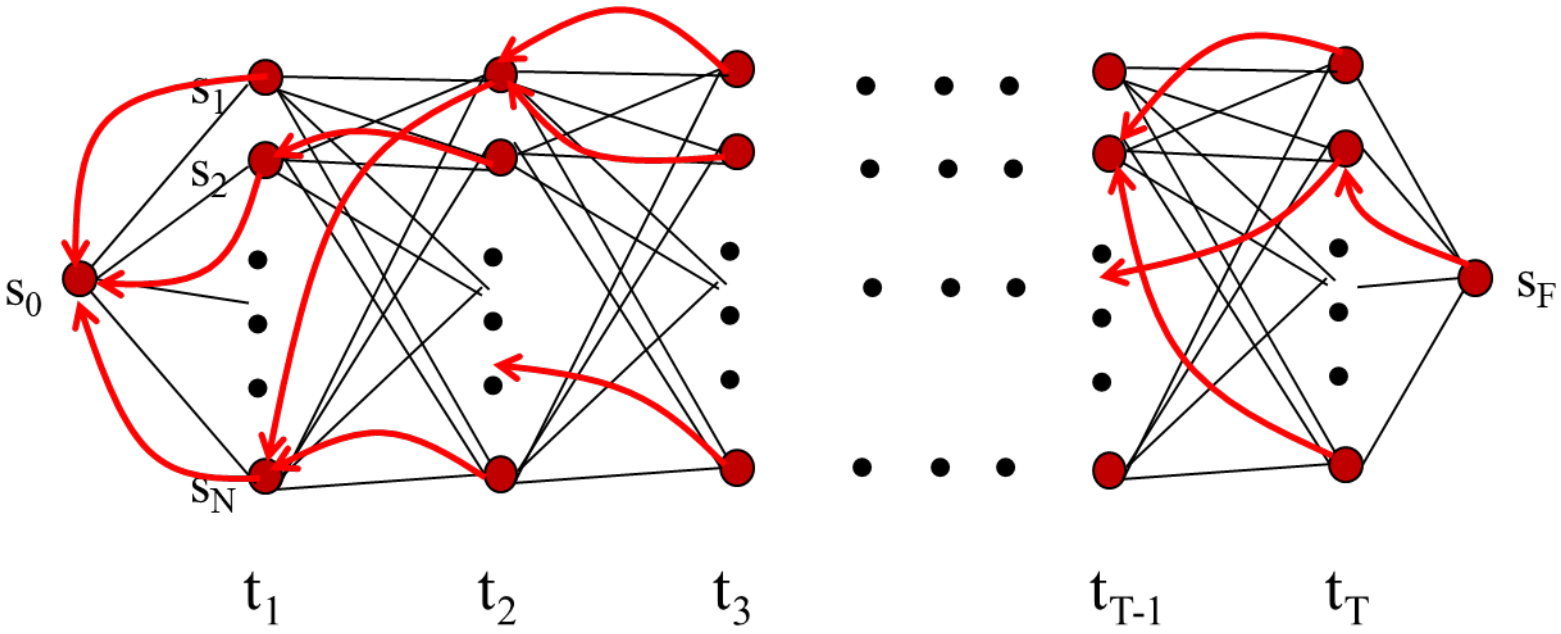

w(1)
w(2)
...
w(i-1)
w(i)
w(i+1)
...
w(N-1)
w(N)
s_1
...
s_j
...
s_T
Tags
Words ("time steps")
word w(i) has tag s_j

$$v_1(j) = a_{0j} b_j(o_1) \quad 1 \le j \le N$$

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i) a_{ij} b_j(o_t) \quad 1 \le j \le N, \; 1 < t \le T$$



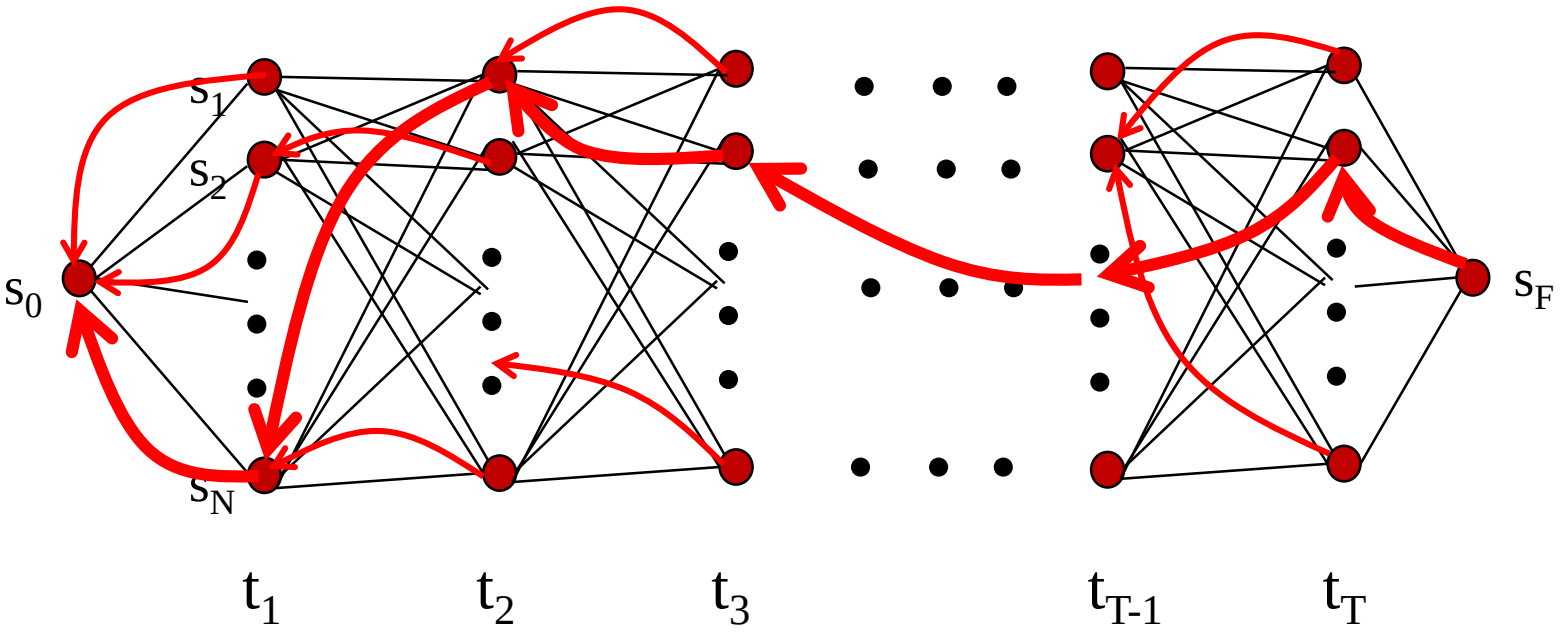
Viterbi Backpointers



$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i) a_{ij} b_j(o_t) \quad 1 \le j \le N, \quad 1 < t \le T$$

$$bt_t(j) = \operatorname{argmax}_{i=1}^{N} v_{t-1}(i) a_{ij} b_j(o_t) \quad 1 \le j \le N, \quad 1 \le t \le T$$

Viterbi Backtrace



Most likely Sequence: $s_0\, s_N\, s_1\, s_2 \ldots s_2\, s_F$