SEQUENCE LABELING FOR PARTS OF SPEECH AND NAMED ENTITIES

Spring 2023

CS6431 Natural Language Processing

Credits

B1: *Speech and Language Processing (Third Edition draft – Jan2022)*

Daniel Jurafsky, James H. Martin

https://www.probabilitycourse.com/chapter8/8_2_3_max_likelihood_estimation.php

https://www.statology.org/likelihood-vs-probability/

https://www.simplilearn.com/tutorials/statistics-tutorial/difference-between-probability-and-likelihood#:~:text=Example%20Scenario,-Suppose%20you%20have&text=However%2C%20when%20calculating%20the%20likelihood,given%20toss%20is%20p%20%3D%200.5.

Assignment

Read:

B1: Chapter 8

Problems:

Part-of-speech (POS) and Named Entity Recognition (NER)

- POS: taking a sequence of words and assigning each word a part of speech like NOUN, VERB, PRONOUN, PREPOSITION, ADVERB, CONJUNCTION, PARTICIPLE, ARTICLE and more.
- NER: assigning words or phrases tags like PERSON, LOCATION, or ORGANIZATION
- Sequence labelling task: POS tagging and NER

Parts of Speech

- Closed Class
  - With relatively fixed membership, e.g., prepositions.
- Open Class
  - New words are continually added. E.g., nouns and verbs (iPhone or to fax)

Noun, verb, adverb, adjective.

	Tag	Description	Example
Open Class	**ADJ**	Adjective: noun modifiers describing properties	*red, young, awesome*
	ADV	Adverb: verb modifiers of time, place, manner	*very, slowly, home, yesterday*
	NOUN	words for persons, places, things, etc.	*algorithm, cat, mango, beauty*
	VERB	words for actions and processes	*draw, provide, go*
	PROPN	Proper noun: name of a person, organization, place, etc..	*Regina, IBM, Colorado*
	INTJ	Interjection: exclamation, greeting, yes/no response, etc.	*oh, um, yes, hello*
Closed Class Words	**ADP**	Adposition (Preposition/Postposition): marks a noun's spacial, temporal, or other relation	*in, on, by, under*
	AUX	Auxiliary: helping verb marking tense, aspect, mood, etc.,	*can, may, should, are*
	CCONJ	Coordinating Conjunction: joins two phrases/clauses	*and, or, but*
	DET	Determiner: marks noun phrase properties	*a, an, the, this*
	NUM	Numeral	*one, two, first, second*
	PART	Particle: a preposition-like form used together with a verb	*up, down, on, off, in, out, at, by*
	PRON	Pronoun: a shorthand for referring to an entity or event	*she, who, I, others*
	SCONJ	Subordinating Conjunction: joins a main clause with a subordinate clause such as a sentential complement	*that, which*
Other	**PUNCT**	Punctuation	; , ()
	SYM	Symbols like $ or emoji	$, %
	X	Other	asdf, qwfg

Universal Dependencies tagset (Nivre et al., 2016a).

Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coord. conj.	*and, but, or*	NNP	proper noun, sing.	*IBM*	TO	"to"	*to*
CD	cardinal number	*one, two*	NNPS	proper noun, plu.	*Carolinas*	UH	interjection	*ah, oops*
DT	determiner	*a, the*	NNS	noun, plural	*llamas*	VB	verb base	*eat*
EX	existential 'there'	*there*	PDT	predeterminer	*all, both*	VBD	verb past tense	*ate*
FW	foreign word	*mea culpa*	POS	possessive ending	*'s*	VBG	verb gerund	*eating*
IN	preposition/ subordin-conj	*of, in, by*	PRP	personal pronoun	*I, you, he*	VBN	verb past partici- ple	*eaten*
JJ	adjective	*yellow*	PRP$	possess. pronoun	*your, one's*	VBP	verb non-3sg-pr	*eat*
JJR	comparative adj	*bigger*	RB	adverb	*quickly*	VBZ	verb 3sg pres	*eats*
JJS	superlative adj	*wildest*	RBR	comparative adv	*faster*	WDT	wh-determ.	*which, that*
LS	list item marker	*1, 2, One*	RBS	superlatv. adv	*fastest*	WP	wh-pronoun	*what, who*
MD	modal	*can, should*	RP	particle	*up, off*	WP$	wh-possess.	*whose*
NN	sing or mass noun	*llama*	SYM	symbol	*+,%, &*	WRB	wh-adverb	*how, where*

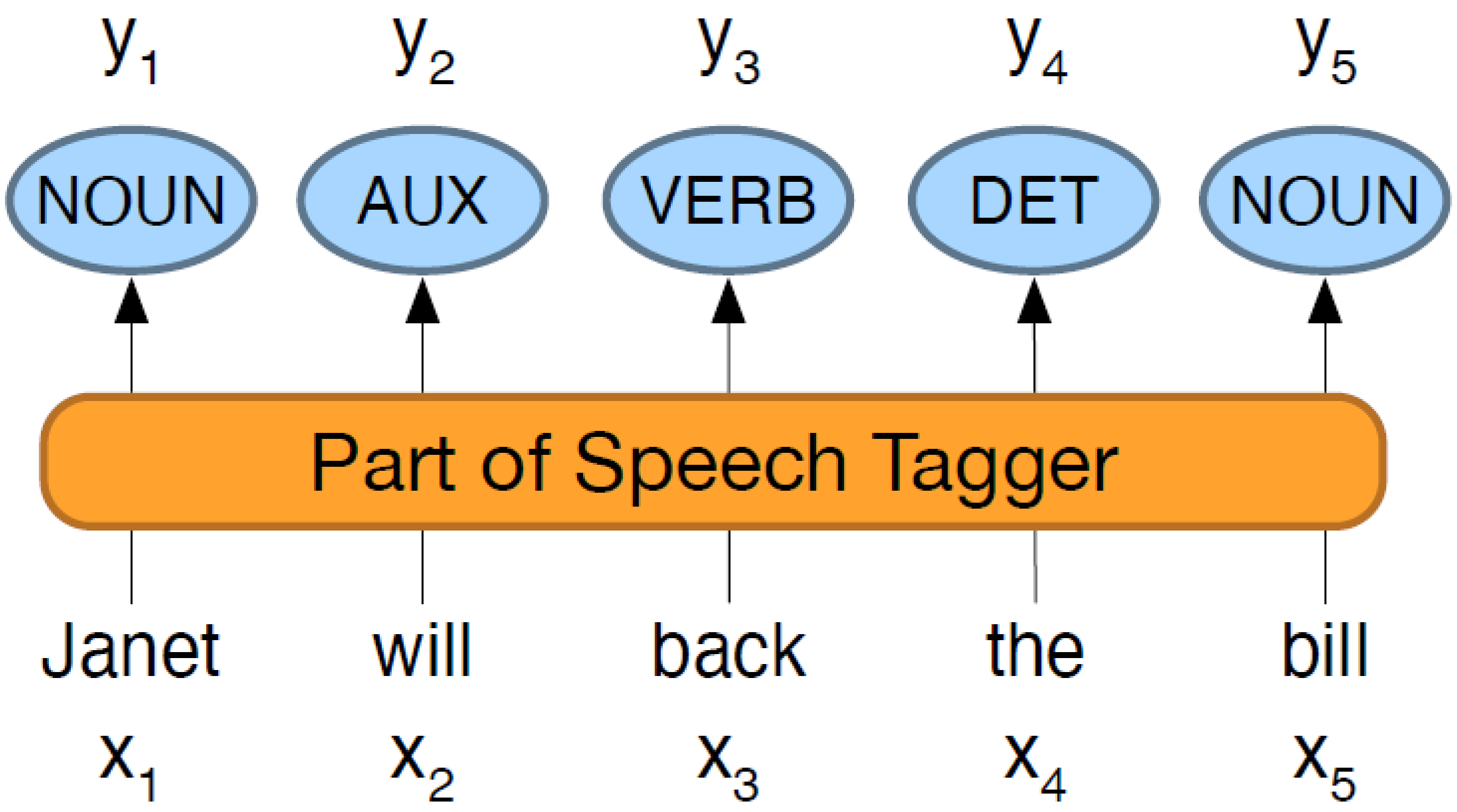
Penn Treebank tagset (Marcus et al., 1993)

- Blue: UD tagset; Red: Penn tagset

There/PRO/EX are/VERB/VBP 70/NUM/CD children/NOUN/NNS there/ADV/RB ./PUNC/.

Preliminary/ADJ/JJ findings/NOUN/NNS were/AUX/VBD reported/VERB/VBN in/ADP/IN today/NOUN/NN 's/PART/POS New/PROPN/NNP England/PROPN/NNP Journal/PROPN/NNP of/ADP/IN Medicine/PROPN/NNP

POS Tagging


y_1 y_2 y_3 y_4 y_5
NOUN
AUX
VERB
DET
NOUN
Part of Speech Tagger
Janet will back the bill
x_1 x_2 x_3 x_4 x_5

- POS tagging is a is a disambiguation task
  - Words are ambiguous; have more than one possible part-of-speech
  - ‘book’ can be a verb (book that flight) or a noun (hand me that book).
  - ‘that’ can be a determiner (Does that flight serve dinner) or a complementizer (I thought that your flight was earlier).
- Very high accuracy for POS tagging algorithms have been achieved
  - 97% (Wu and Dredze, 2019), (Manning, 2011).
    - The same as what humans can achieve

- Amount of ambiguity

Types:		WSJ		Brown	
Unambiguous	(1 tag)	44,432	(**86%**)	45,799	(**85%**)
Ambiguous	(2+ tags)	7,025	(**14%**)	8,050	(**15%**)
Tokens:					
Unambiguous	(1 tag)	577,421	(**45%**)	384,349	(**33%**)
Ambiguous	(2+ tags)	711,780	(**55%**)	786,646	(**67%**)

- Ambiguous words though are only 14-15% of the vocabulary, are very common, and 55-67% of word tokens in running text are ambiguous

A Baseline Classifier for Ambiguous Words

- Always compare a classifier against a baseline at least as good as the most frequent class baseline (assigning each token to the class it occurred in most often in the training set).
  - Produces an accuracy of 92% (just 5% less than human standard)

Named Entities and Named Entity Tagging

- Named Entity – roughly speaking, anything that can be referred to with a proper name: a person, a location, an organization.
  1. PER (person)
  2. LOC (location)
  3. ORG (organization)
  4. GPE (geo-political entity)

Type	Tag	Sample Categories	Example sentences
People	PER	people, characters	**Turing** is a giant of computer science.
Organization	ORG	companies, sports teams	The **IPCC** warned about the cyclone.
Location	LOC	regions, mountains, seas	**Mt. Sanitas** is in **Sunshine Canyon**.
Geo-Political Entity	GPE	countries, states	**Palo Alto** is raising the fees for parking.

- A sample output from an NER tagger

Citing high fuel prices, [$_{\text{ORG}}$ **United Airlines**] said [$_{\text{TIME}}$ **Friday**] it has increased fares by [$_{\text{MONEY}}$ **$6**] per round trip on flights to some cities also served by lower-cost carriers. [$_{\text{ORG}}$ **American Airlines**], a unit of [$_{\text{ORG}}$ **AMR Corp.**], immediately matched the move, spokesman [$_{\text{PER}}$ **Tim Wagner**] said. [$_{\text{ORG}}$ **United**], a unit of [$_{\text{ORG}}$ **UAL Corp.**], said the increase took effect [$_{\text{TIME}}$ **Thursday**] and applies to most routes where it competes against discount carriers, such as [$_{\text{LOC}}$ **Chicago**] to [$_{\text{LOC}}$ **Dallas**] and [$_{\text{LOC}}$ **Denver**] to [$_{\text{LOC}}$ **San Francisco**].

- Issues
  - Segmentation problem in NER – NEs can span words
    - BIO tagging (Ramshaw and Marcus, 1995).

      solution of NER
  - Type ambiguity

[$_{PER}$ Washington] was born into slavery on the farm of James Burroughs.
[$_{ORG}$ Washington] went up 2 games to 1 in the four-game series.
Blair arrived in [$_{LOC}$ Washington] for what may well be his last state visit.
In June, [$_{GPE}$ Washington] passed a primary seatbelt law.

BIO Tagging

Write on desk

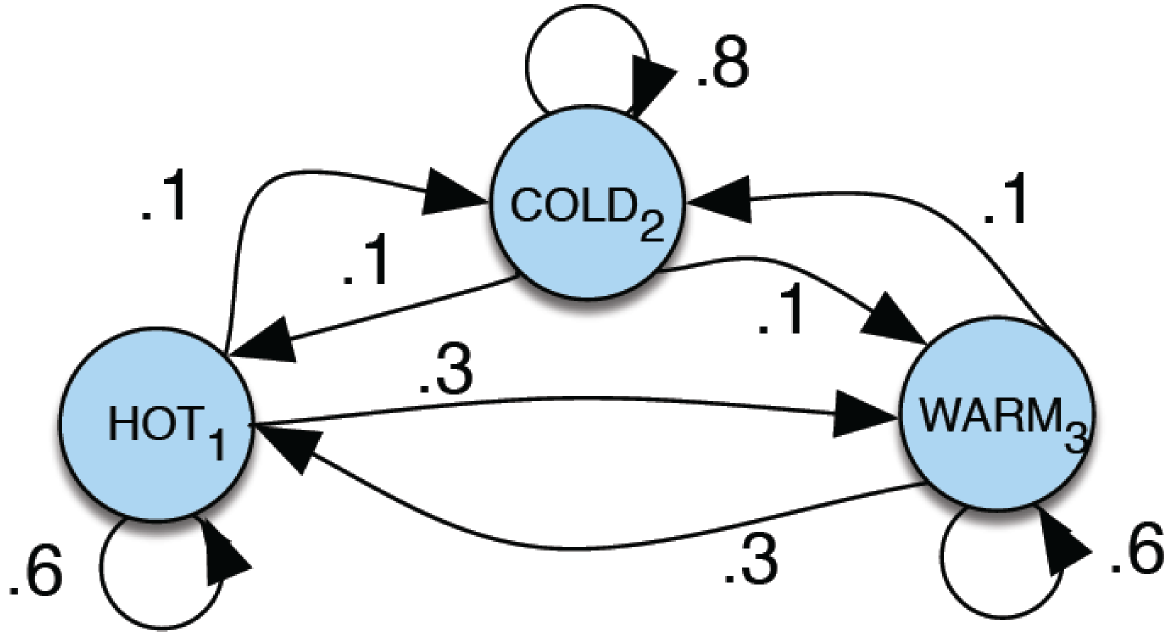
- Captures both boundary and NE type

Words	IO Label	BIO Label	BIOES Label
Jane	I-PER	B-PER	B-PER
Villanueva	I-PER	I-PER	E-PER
of	O	O	O
United	I-ORG	B-ORG	B-ORG
Airlines	I-ORG	I-ORG	I-ORG
Holding	I-ORG	I-ORG	E-ORG
discussed	O	O	O
the	O	O	O
Chicago	I-LOC	B-LOC	S-LOC
route	O	O	O
.	O	O	O

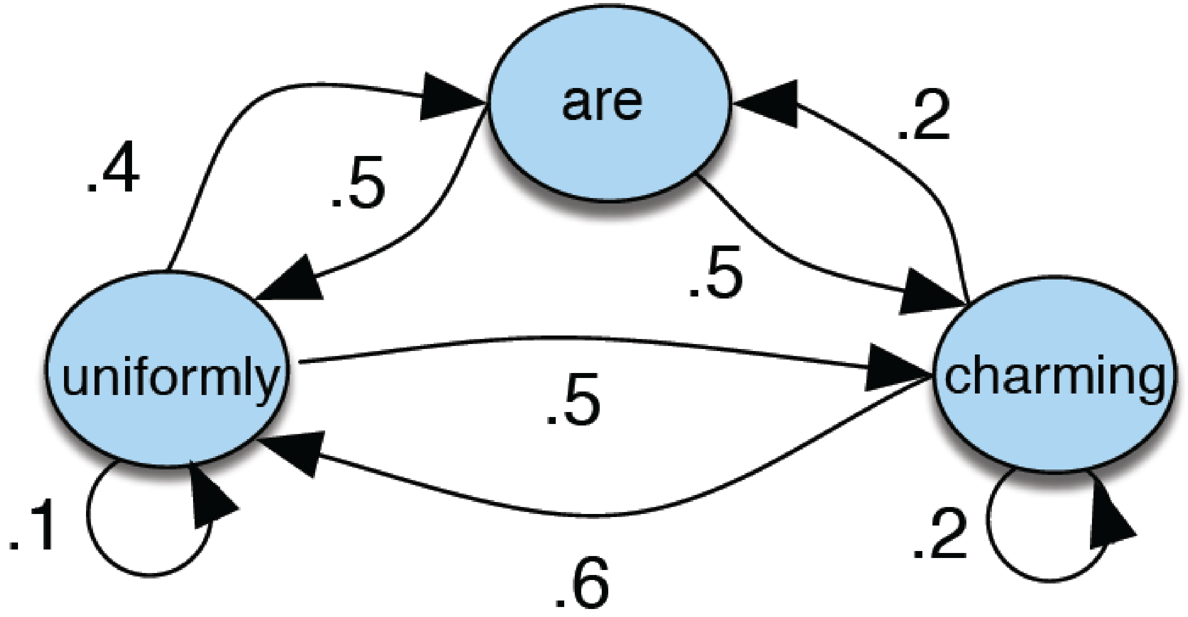
HMM Part-of-Speech Tagging

Markov Chain

- Consists of states and probability sequence
- Very strong assumption: knowledge of current state is sufficient to predict the next state



(a)



(b)

- Formally, a Markov Chain is defined as

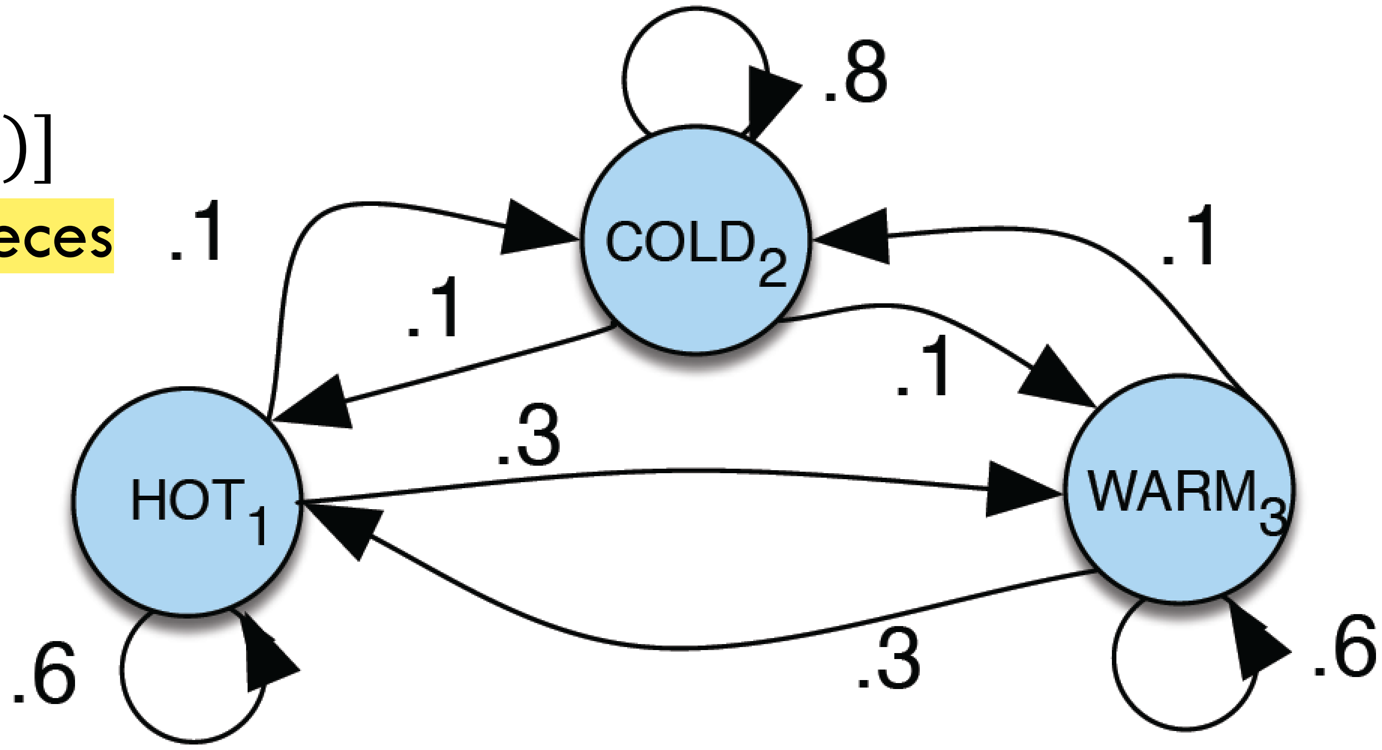
$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} a_{12} \ldots a_{N1} \ldots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

Take the initial probabilities

$$\pi = [.1\ (\text{cold}), .7(hot), .2(\text{warm})]$$

Compute the probabilities for seuqneces

1. hot hot hot hot
2. cold hot cold hot



Hidden Markov Model

- Sometimes the events we are interested in are ***hidden***: we don't observe them directly.
  - E.g., POS tags in a sequence of words
- An HMM allows to model both
  - Observed events (like words that we see in the input)
  - Hidden events (like part-of-speech tags)

- Formally, an HMM is defined as

$Q = q_1 q_2 \dots q_N$	a set of N **states**
$A = a_{11} \dots a_{ij} \dots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \dots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, \dots, v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state q_i
$\pi = \pi_1, \pi_2, \dots, \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

- Simplifying assumptions
- One: **Markov Assumption:** $P(q_i|q_1,...,q_{i-1}) = P(q_i|q_{i-1})$ current state depends only on previous state
- Two: the probability of an output observation o_i depends only on the state q_i that produced the observation and not on any other states or any other observations

Output Independence: $P(o_i|q_1,\ldots q_i,\ldots,q_T,o_1,\ldots,o_i,\ldots,o_T) = P(o_i|q_i)$

The components of an HMM tagger

- w_i: denotes a word (e.g., 'will')
- t_i: tag associated with w_i (e.g., 'MD' stands for modal verb)
- A probability matrix: contains $P(t_i|t_{i-1})$ - probability of a tag occurring given the previous tag
  - E.g., in "…will race…", tag 'MD' (modal verb) is followed by 'VB' (verb in the base f
  - Maximum likelihood estimate

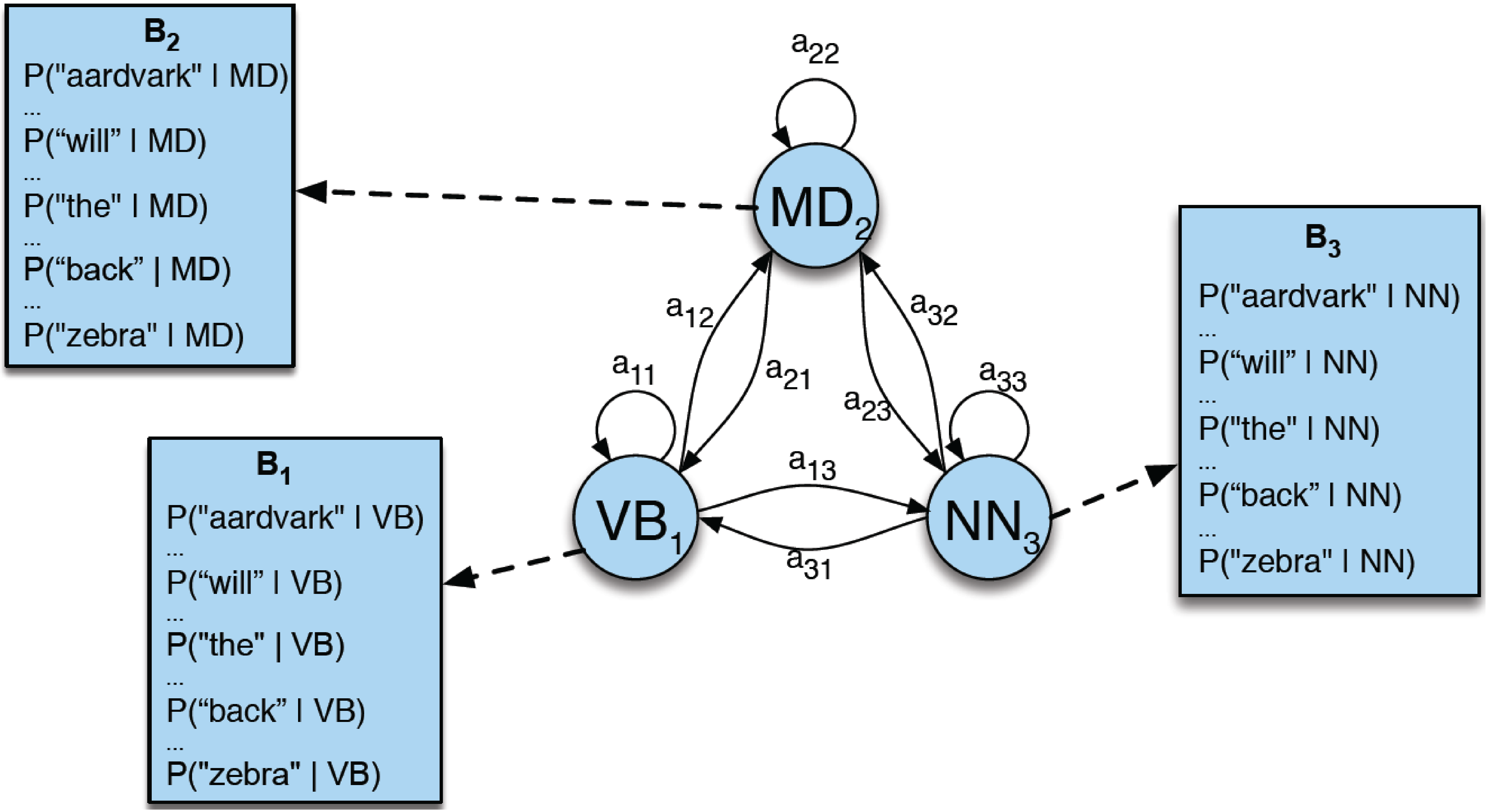
$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

Write on desk

- B emission probability matrix: $P(w_i|t_i)$ - represent the probability, given a tag (say 'MD') will be associated with a given word (say 'will')
  - Maximum likelihood estimate

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

- Note: Both A and B can be pre-computed from a corpus


B2
P("aardvark" | MD)
...
P("will" | MD)
...
P("the" | MD)
...
P("back" | MD)
...
P("zebra" | MD)
a22
MD2
a12
a21
a32
a23
a11
a33
VB1
a13
a31
NN3
B1
P("aardvark" | VB)
...
P("will" | VB)
...
P("the" | VB)
...
P("back" | VB)
...
P("zebra" | VB)
B3
P("aardvark" | NN)
...
P("will" | NN)
...
P("the" | NN)
...
P("back" | NN)
...
P("zebra" | NN)

HMM tagging as decoding

- The task of determining the hidden variables sequence corresponding to the sequence of observations is called decoding

Decoding: Given as input an HMM $\lambda = (A, B)$ and a sequence of observations $O = o_1, o_2, \ldots, o_T$, find the most probable sequence of states $Q = q_1 q_2 q_3 \ldots q_T$.

- For part-of-speech tagging, the goal of HMM decoding is to choose the tag sequence $t_1 \ldots t_n$ that is most probable given the observation sequence of n words $w_1 \ldots w_n$

$$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{\operatorname{argmax}} P(t_1 \ldots t_n | w_1 \ldots w_n)$$

$$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{\operatorname{argmax}} P(t_1 \ldots t_n | w_1 \ldots w_n)$$

- We will use Bayes' rule to instead compute

$$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{\operatorname{argmax}} \frac{P(w_1 \ldots w_n | t_1 \ldots t_n) P(t_1 \ldots t_n)}{P(w_1 \ldots w_n)}$$

- Dropping the denominator

$$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{\operatorname{argmax}} P(w_1 \ldots w_n | t_1 \ldots t_n) P(t_1 \ldots t_n)$$

$$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{\operatorname{argmax}} P(w_1 \ldots w_n | t_1 \ldots t_n) P(t_1 \ldots t_n)$$

- Assumption 1: the probability of a word appearing depends only on its own tag and is independent of neighboring words and tags

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

- Assumption 2: Bigram assumption: the probability of a tag is dependent only on the previous tag, rather than the entire tag sequence

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

$$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{\operatorname{argmax}} P(t_1 \ldots t_n | w_1 \ldots w_n) \approx \underset{t_1 \ldots t_n}{\operatorname{argmax}} \prod_{i=1}^{n} \overbrace{P(w_i | t_i)}^{\text{emission}} \overbrace{P(t_i | t_{i-1})}^{\text{transition}}$$

The Viterbi Algorithm – a dynamic programming solution

function VITERBI(*observations* of len *T*,*state-graph* of len *N*) **returns** *best-path, path-prob*

create a path probability matrix *viterbi[N,T]*

for each state *s* **from** 1 **to** *N* **do** ; initialization step

$$viterbi[s,1] \leftarrow \pi_s * b_s(o_1)$$

$$backpointer[s,1] \leftarrow 0$$

Write on desk

for each time step *t* **from** 2 **to** *T* **do** ; recursion step

for each state *s* **from** 1 **to** *N* **do**

$$viterbi[s,t] \leftarrow \max_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$$

$$backpointer[s,t] \leftarrow \operatorname{argmax}_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$$

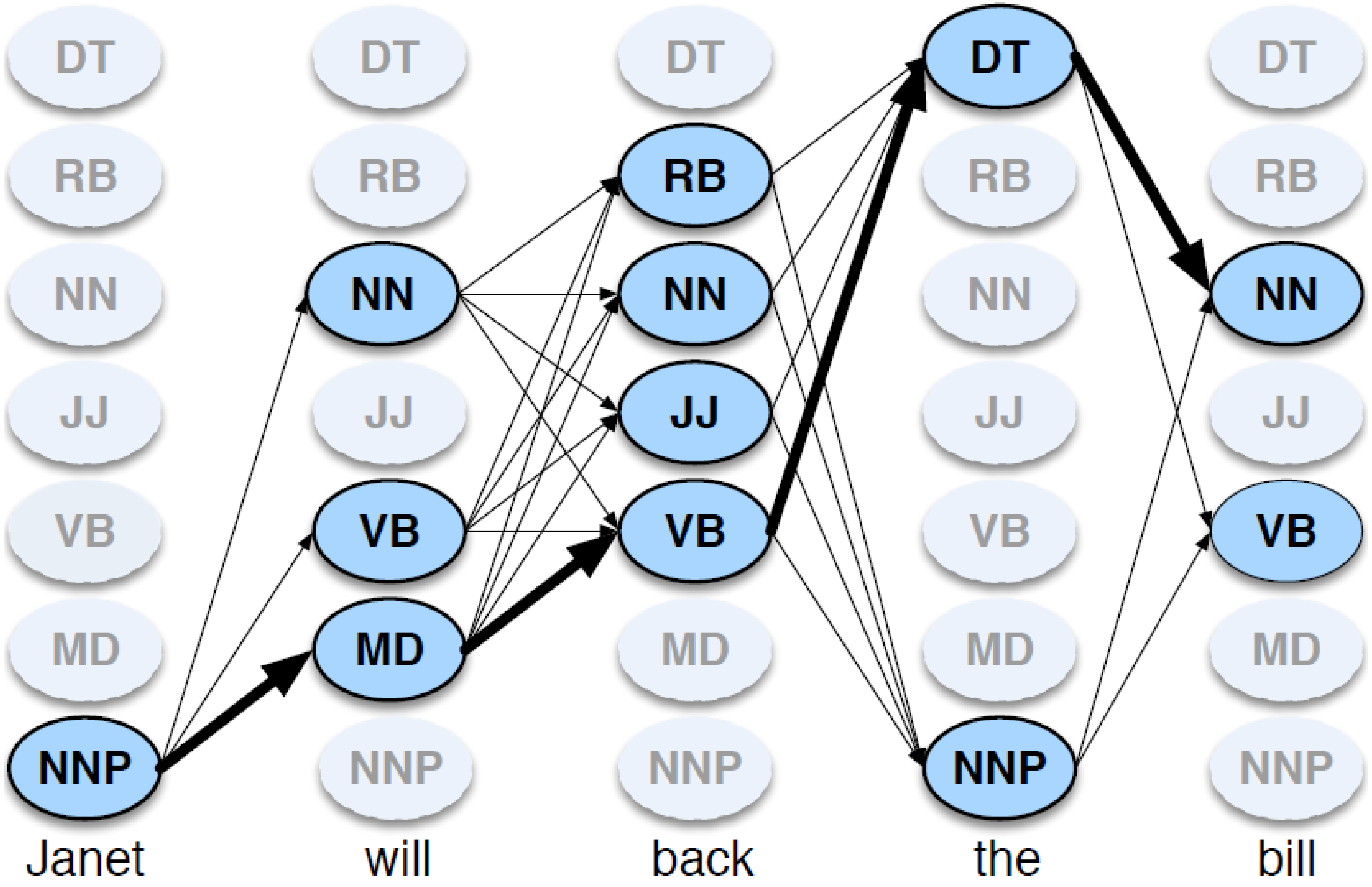
$$bestpathprob \leftarrow \max_{s=1}^{N} viterbi[s,T]$$; termination step

$$bestpathpointer \leftarrow \operatorname{argmax}_{s=1}^{N} viterbi[s,T]$$; termination step

bestpath ← the path starting at state *bestpathpointer*, that follows backpointer[] to states back in time

return *bestpath*, *bestpathprob*

- $viterbi[N,T]$ or $v[N,T]$: with one column for each observation o_t and one row for each state q_i in the state graph.

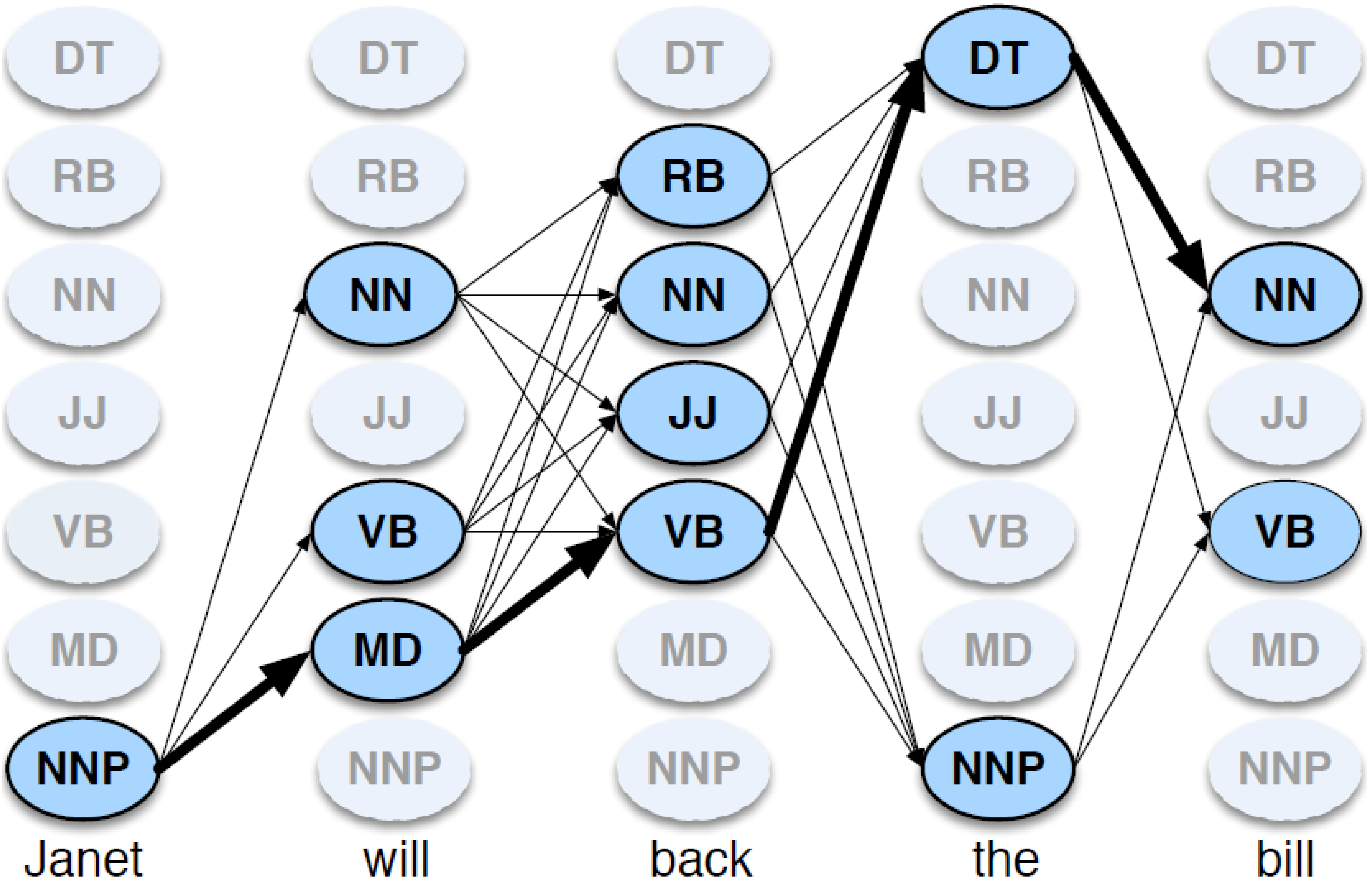


DT - determiner,
RB - adverb,
NN – sing or mass noun,
JJ – adjective
VB – verb base
MD – modal verb
NNP – proper noun, sing.

- $v_t(j)$: represents the probability that the HMM is in state j after seeing the first t observations and passing through the most probable state sequence $q_1, \ldots, q_{t-1}$, given the HMM λ.

- $v_t(j)$: is computed by recursively taking the most probable path that could lead us to this cell
- Dynamic programming way

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

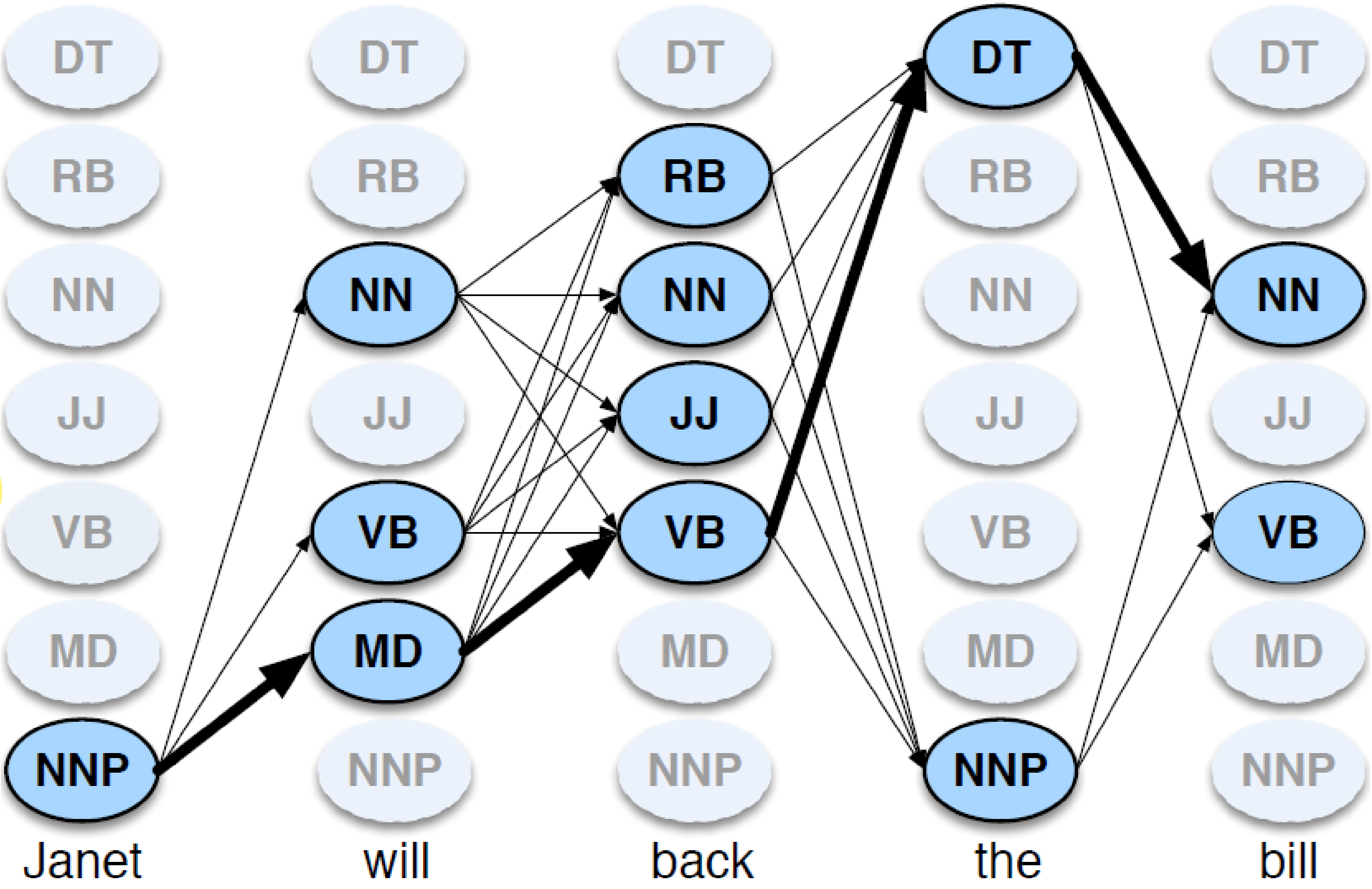


$v_{t-1}(i)$	the **previous Viterbi path probability** from the previous time step
a_{ij}	the **transition probability** from previous state q_i to current state q_j
$b_j(o_t)$	the **state observation likelihood** of the observation symbol o_t given the current state j

- Dynamic programming way

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

write on desk

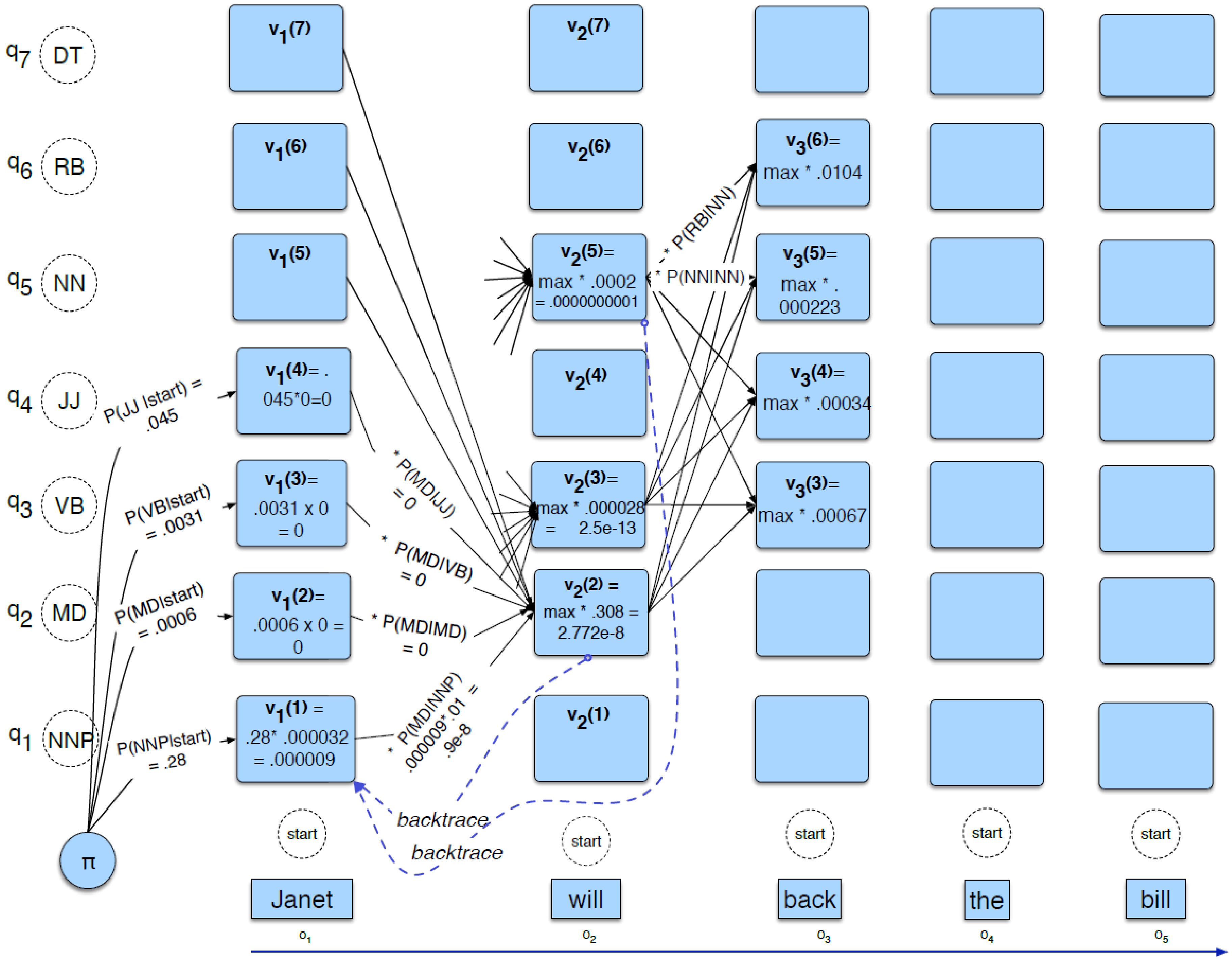


- Let us tag "Janet will back the bill"
  - Correct sol: Janet/NNP will/MD back/VB the/DT bill/NN
- A: Transition probabilities $P(t_i|t_{i-1})$ – precomputed from corpus

	NNP	**MD**	**VB**	**JJ**	**NN**	**RB**	**DT**
$<s>$	0.2767	0.0006	0.0031	0.0453	0.0449	0.0510	0.2026
NNP	0.3777	0.0110	0.0009	0.0084	0.0584	0.0090	0.0025
MD	0.0008	0.0002	0.7968	0.0005	0.0008	0.1698	0.0041
VB	0.0322	0.0005	0.0050	0.0837	0.0615	0.0514	0.2231
JJ	0.0366	0.0004	0.0001	0.0733	0.4509	0.0036	0.0036
NN	0.0096	0.0176	0.0014	0.0086	0.1216	0.0177	0.0068
RB	0.0068	0.0102	0.1011	0.1012	0.0120	0.0728	0.0479
DT	0.1147	0.0021	0.0002	0.2157	0.4744	0.0102	0.0017

- Observation likelihoods/ emission probabilities: B: $b_i(o_t)$ or $p(w_i|t_i)$

	Janet	will	back	the	bill
NNP	0.000032	0	0	0.000048	0
MD	0	0.308431	0	0	0
VB	0	0.000028	0.000672	0	0.000028
JJ	0	0	0.000340	0	0
NN	0	0.000200	0.000223	0	0.002337
RB	0	0	0.010446	0	0
DT	0	0	0	0.506099	0



q7 DT
q6 RB
q5 NN
q4 JJ
q3 VB
q2 MD
q1 NNP
π
v1(7)
v1(6)
v1(5)
v1(4)= .045*0=0
v1(3)= .0031 x 0 = 0
v1(2)= .0006 x 0 = 0
v1(1) = .28* .000032 = .000009
P(JJ |start) = .045
P(VB|start) = .0031
P(MD|start) = .0006
P(NNP|start) = .28
* P(MD|JJ) = 0
* P(MD|VB) = 0
* P(MD|MD) = 0
* P(MD|NNP) .000009*.01 = .9e-8
v2(7)
v2(6)
v2(5)= max * .0002 = .0000000001
v2(4)
v2(3)= max * .000028 = 2.5e-13
v2(2) = max * .308 = 2.772e-8
v2(1)
* P(RB|NN)
* P(NN|NN)
v3(6)= max * .0104
v3(5)= max * . 000223
v3(4)= max * .00034
v3(3)= max * .00067
backtrace
backtrace
start
start
start
start
start
Janet
will
back
the
bill
o1
o2
o3
o4
o5