HMM Tagging as Decoding

- HMM taggers make two simplifying assumptions.
- **The first is that the probability of a word appearing depends only on its own tag and is independent of neighboring words and tags:**

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

- **The second assumption, the bigram assumption (first-order HMM), is that the probability of a tag is dependent only on the previous tag, rather than the entire tag sequence:**

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

HMM Tagging as Decoding

- Plugging the simplifying assumptions results in the following equation for **the most probable tag sequence from a bigram tagger (first-order HMM)**:

$$\hat{\mathbf{T}} = \underset{\mathbf{T} \in \tau}{\mathbf{argmax}}\ \mathbf{P}(\mathbf{T}|\mathbf{W}) = \underset{\mathbf{T} \in \tau}{\mathbf{argmax}} \prod_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \mathbf{P}(\mathbf{w_i}|\mathbf{t_i})\, \mathbf{P}(\mathbf{t_i}|\mathbf{t_{i-1}})$$

emission probability

transition probability

Viterbi Algorithm for POS Tagging Example: Janet will back the bill

- **Observation likelihoods B** computed from the WSJ corpus without smoothing

	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

Viterbi Algorithm for POS Tagging Example: Janet will back the bill

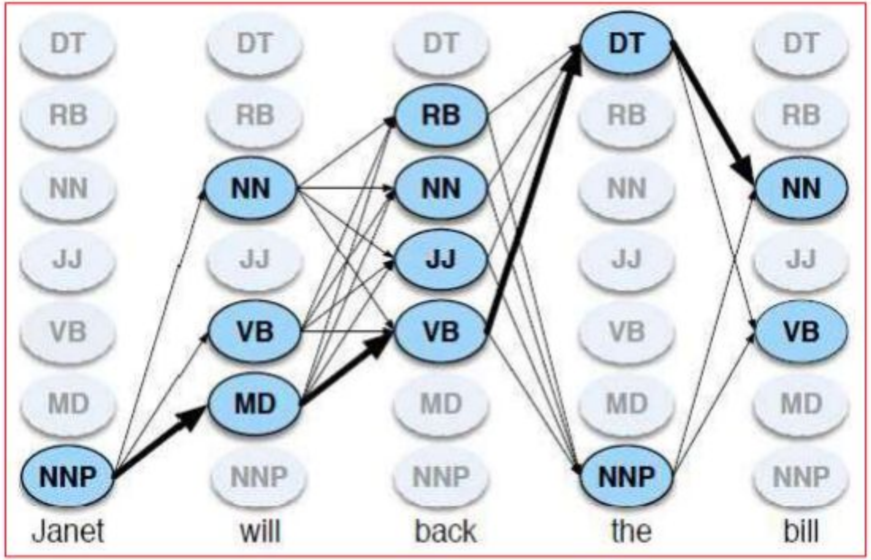
- The **A transition probabilities $P(t_i|t_{i-1})$** computed from the WSJ corpus without smoothing.

	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

Viterbi Algorithm for POS Tagging Example: Janet will back the bill

Sketch of Viterbi matrix for **Janet will back the bill**,

– possible tags for each word and highlighting the path corresponding to the correct tag sequence through the hidden states. States (parts-of-speech) which have a zero probability of generating a particular word according to the B matrix (such as the probability that a determiner DT will be realized as Janet) are greyed out..

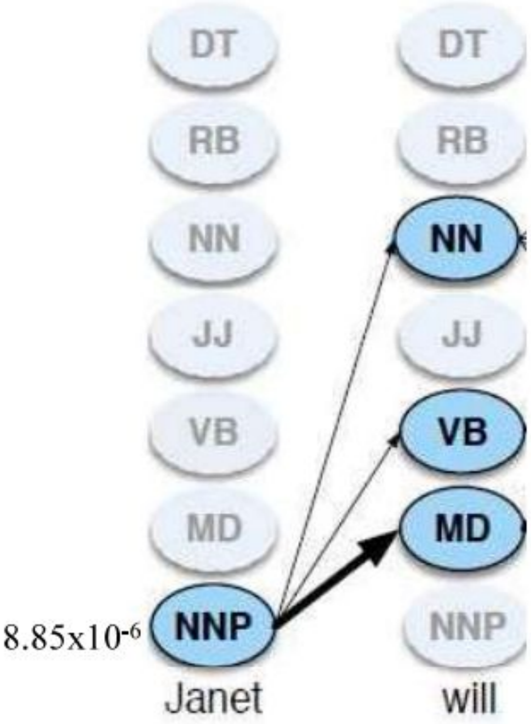


Viterbi Algorithm for POS Tagging Example: Janet will back the bill

Viterbi[NNP,Janet] =
P(NNP|<s>)*P(Janet|NNP) = 0.2767*0.000032 = 0.00000885 = 8.85×10^{-6}

Viterbi Algorithm for POS Tagging

Example: Janet will back the bill

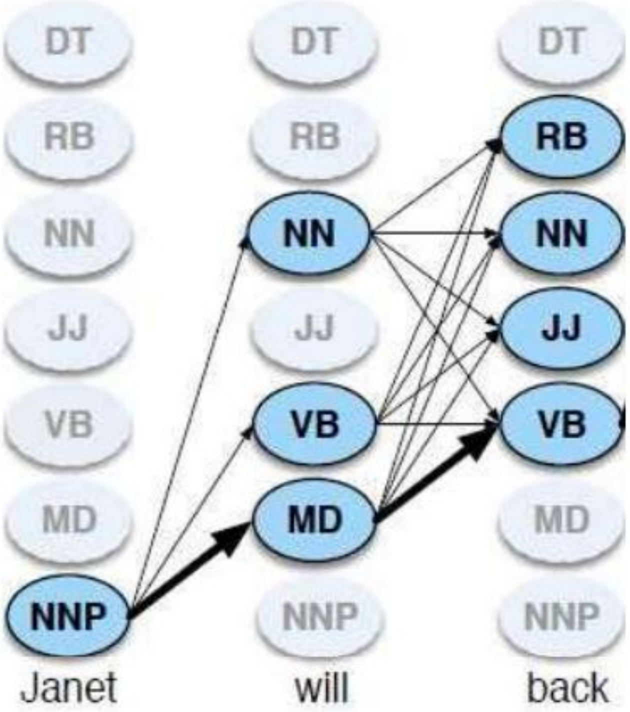


Viterbi[NN,will] = Viterbi[NNP,Janet]*P(NN|NNP)*P(will|NN)
$= 8.85\text{x}10^{-6}*0.0584*0.0002 = 1.03368\text{x}10^{-9}$

Viterbi[VB,will] = Viterbi[NNP,Janet]*P(VB|NNP)*P(will|VB)
$= 8.85\text{x}10^{-6}*0.0009*0.000028 = 0.22302\text{x}10^{-12}$

Viterbi[MD,will] = Viterbi[NNP,Janet]*P(MD|NNP)*P(will|MD)
$= 8.85\text{x}10^{-6}*0.011*0.308431 = 0.203\text{x}10^{-6}$

Viterbi Algorithm for POS Tagging
Example: Janet will back the bill



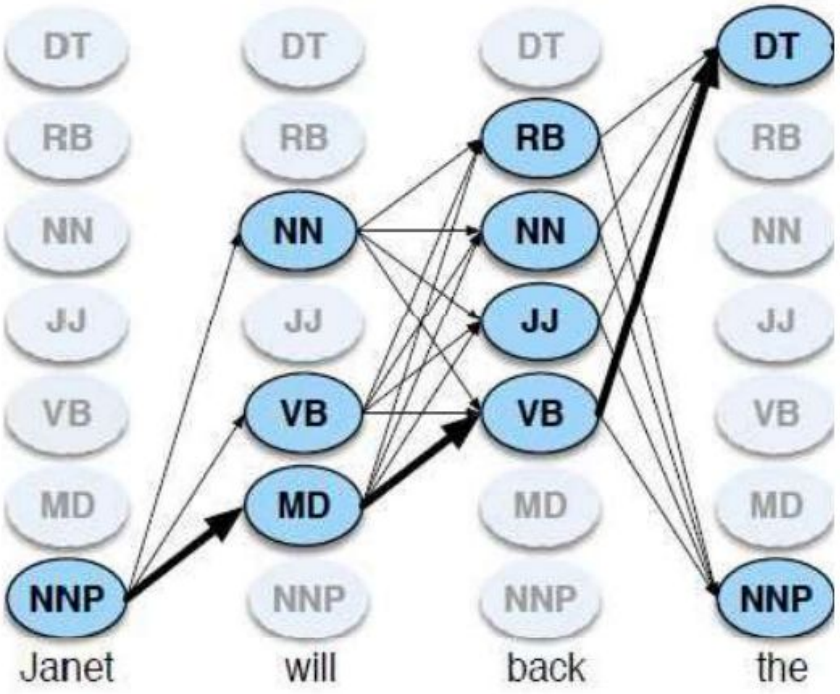
Viterbi[RB,back]=max({Viterbi[NN,will]*P(RB|NN)*P(back|RB),
Viterbi[VB,will]*P(RB|VB)*P(back|RB),
Viterbi[MD,will]*P(RB|MD)*P(back|RB)})

Viterbi[NN,back]=max({Viterbi[NN,will]*P(NN|NN)*P(back|NN),
Viterbi[VB,will]*P(NN|VB)*P(back|NN),
Viterbi[MD,will]*P(NN|MD)*P(back|NN)})

Viterbi[JJ,back]=max({Viterbi[NN,will]*P(JJ|NN)*P(back|JJ),
Viterbi[VB,will]*P(JJ|VB)*P(back|JJ),
Viterbi[MD,will]*P(JJ|MD)*P(back|JJ)})

Viterbi[VB,back]=max({Viterbi[NN,will]*P(VB|NN)*P(back|VB),
Viterbi[VB,will]*P(VB|VB)*P(back|VB),
Viterbi[MD,will]*P(VB|MD)*P(back|VB)})

Viterbi Algorithm for POS Tagging
Example: Janet will back the bill

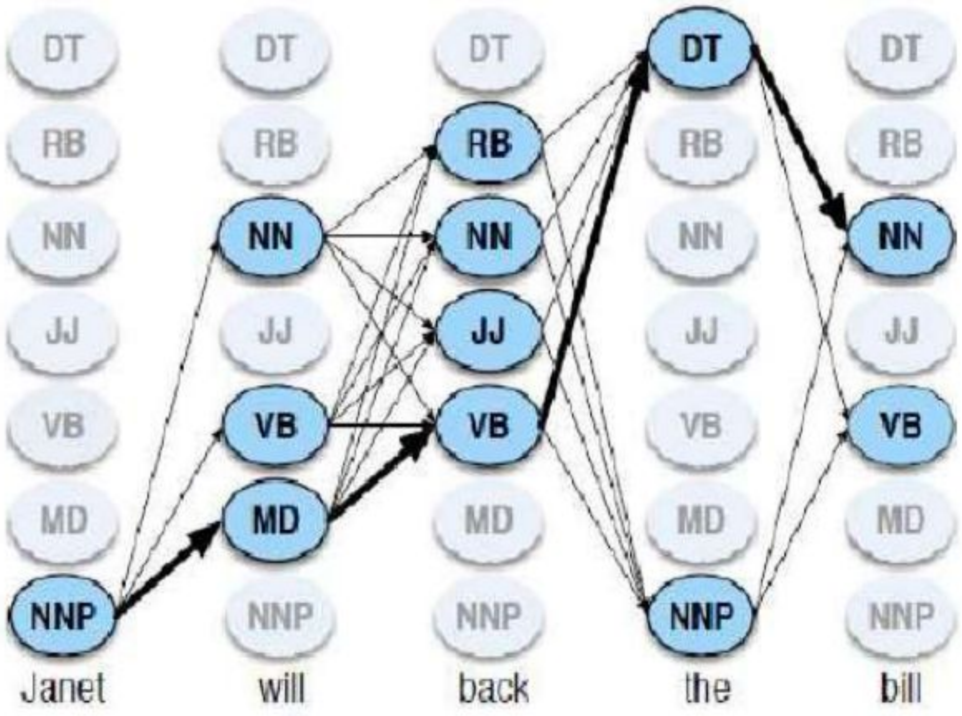


Viterbi[DT,the]=
max({Viterbi[RB,back]*P(DT|RB)*P(the|DT),
Viterbi[NN,back]*P(DT|NN)*P(the|DT),
Viterbi[JJ,back]*P(DT|JJ)*P(the|DT),
Viterbi[VB,back]*P(DT|VB)*P(the|DT)})

Viterbi[NNP,the]=
max({Viterbi[RB,back]*P(NNP|RB)*P(the|NNP),
Viterbi[NN,back]*P(NNP|NN)*P(the|NNP),
Viterbi[JJ,back]*P(NNP|JJ)*P(the|NNP),
Viterbi[VB,back]*P(NNP|VB)*P(the|NNP)})

Viterbi Algorithm for POS Tagging

Example: Janet will back the bill

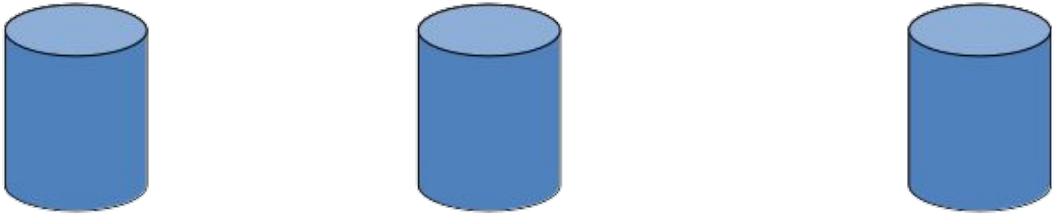


Viterbi[NN,bill]=
max({Viterbi[DT,the]*P(NN|DT)*P(bill|NN),
Viterbi[NNP,the]*P(NN|NNP)*P(bill|NN)})

Viterbi[VB,bill]=
max({Viterbi[DT,the]*P(VB|DT)*P(bill|VB),
Viterbi[NNP,the]*P(VB|NNP)*P(bill|VB)})

Another Example

A colored ball choosing example :



Urn 1
# of Red = 30
# of Green = 50
# of Blue = 20

Urn 2
# of Red = 10
# of Green = 40
# of Blue = 50

Urn 3
# of Red =60
# of Green =10
# of Blue = 30

Probability of transition to another Urn after picking a ball:

	U1	U2	U3
U1	0.1	0.4	0.5
U2	0.6	0.2	0.2
U3	0.3	0.4	0.3

Example (contd.)

Given :

	U1	U2	U3
U1	0.1	0.4	0.5
U2	0.6	0.2	0.2
U3	0.3	0.4	0.3

and

	R	G	B
U1	0.3	0.5	0.2
U2	0.1	0.4	0.5
U3	0.6	0.1	0.3

Observation : RRGGBRGR

State Sequence : ??

Not so Easily Computable.

	u1	u2	u3
<s>	0.4	0.3	.3
u1	0.1	0.4	.5
u2	.6	.2	.2
u3	.3	.4	.3