CS598JHM: **Advanced NLP** (Spring ’10)

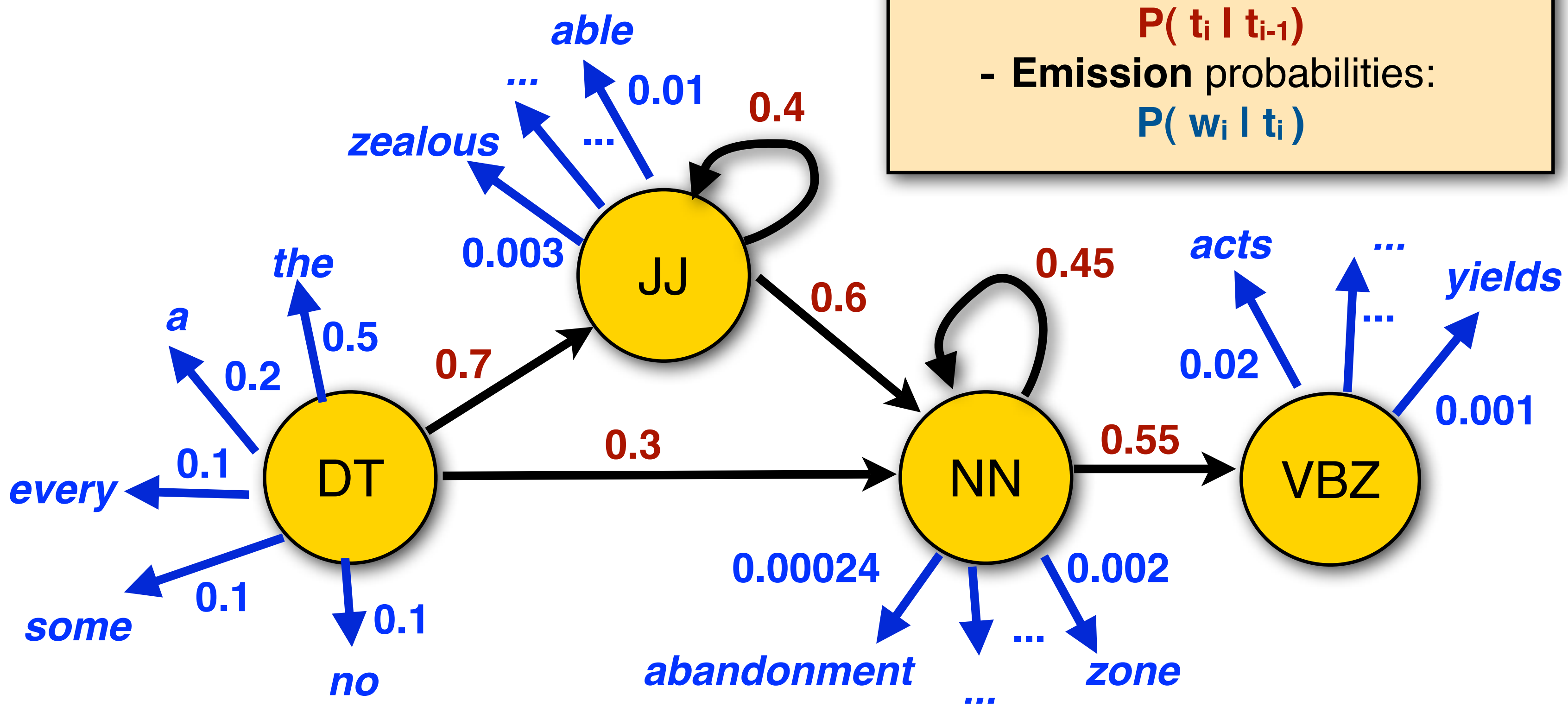
Forward/Backward

Julia Hockenmaier
juliahmr@illinois.edu
3324 Siebel Center

http://www.cs.uiuc.edu/class/sp10/cs598jhm

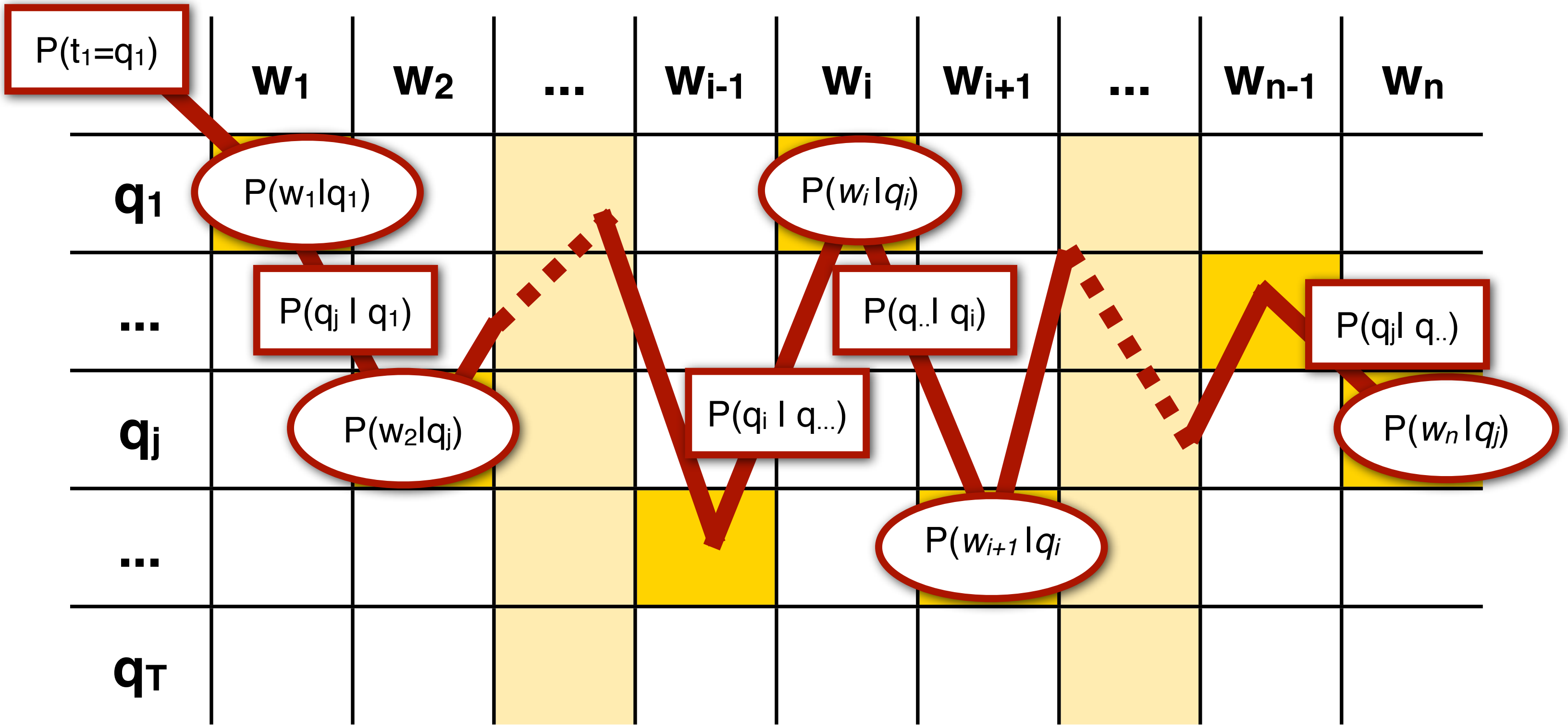
HMMs as probabilistic automata

An HMM defines
- **Transition** probabilities: $P(\, t_i \mid t_{i-1})$
- **Emission** probabilities: $P(\, w_i \mid t_i \,)$



The Forward algorithm

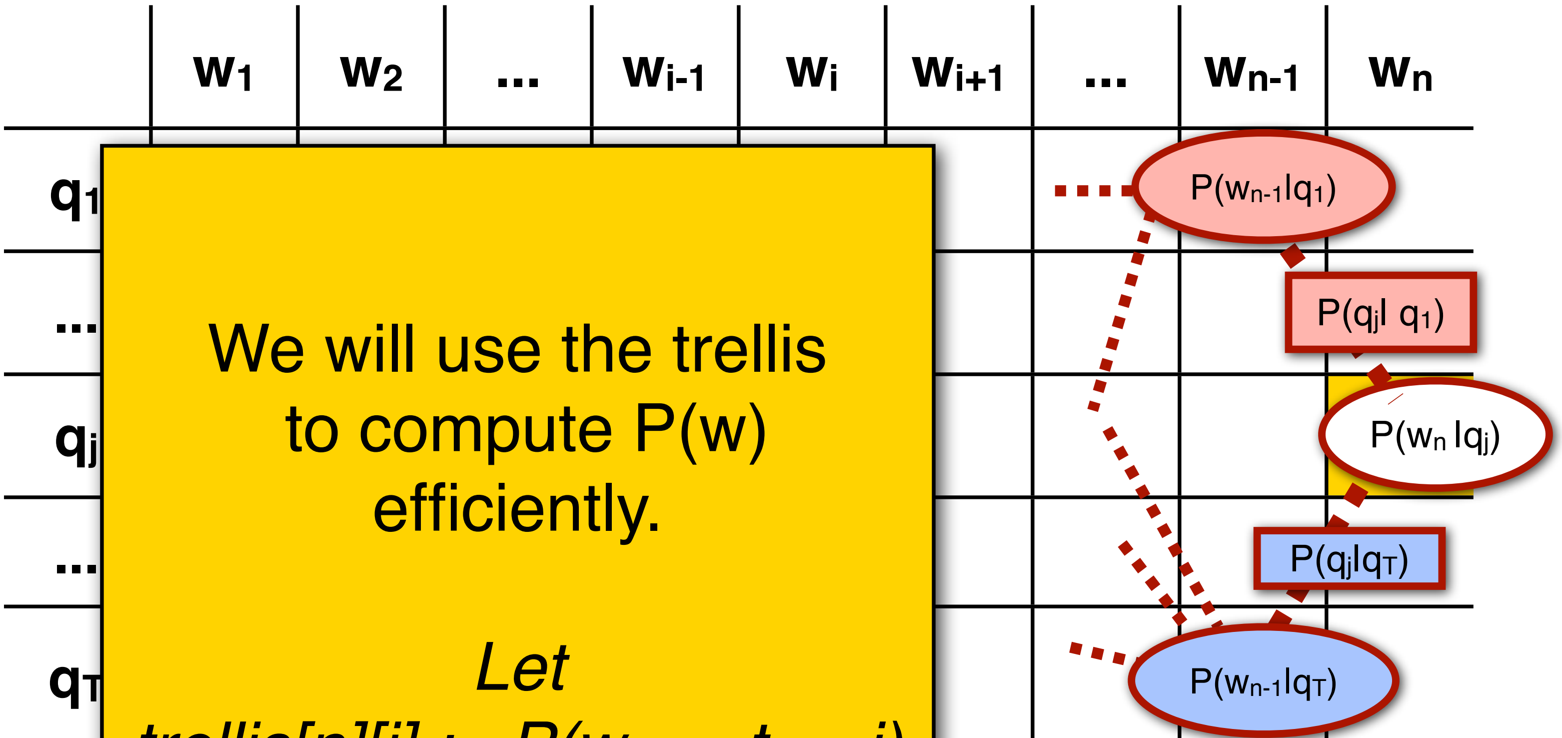
Computing P(t,w)



- One path through the trellis = one tag sequence
- We just multiply the probabilities

$$P(\mathbf{t}, \mathbf{w}) \quad = \quad P(t_1)P(w_1|t_1)\prod_{i=2}^{T} P(t_i|t_{i-1})P(w_i|t_i)$$

2. Finding P(w) = ∑t P(t,w)



We will use the trellis to compute P(w) efficiently.

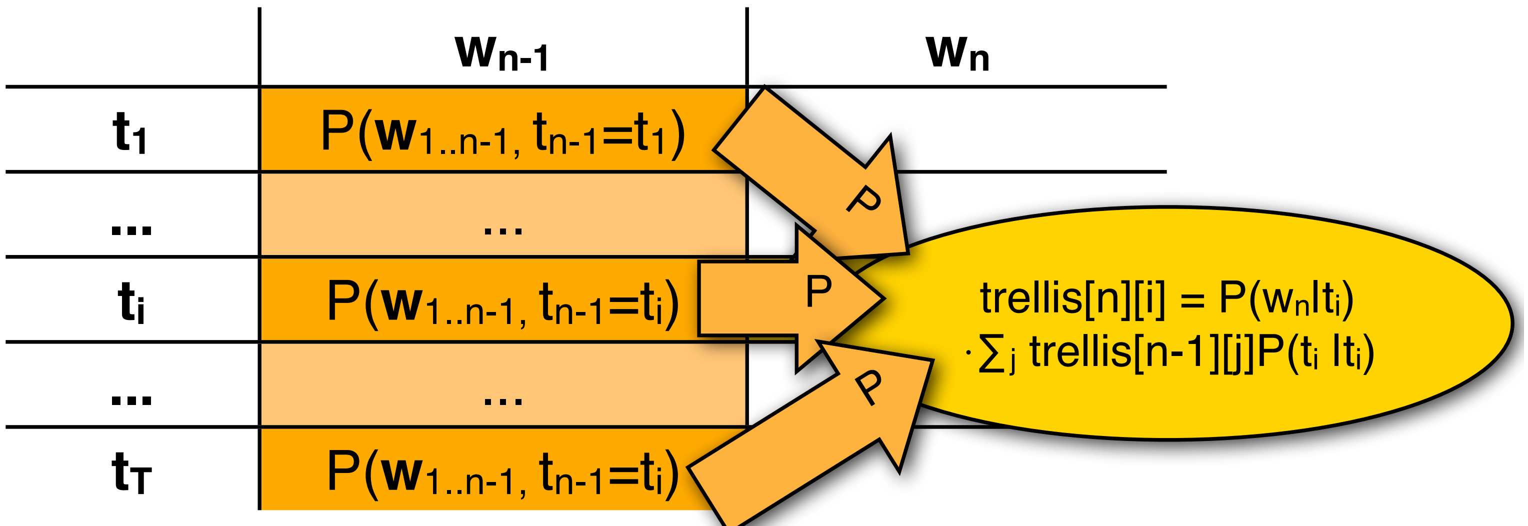
Let

trellis[n][i] := $P(w_{1..n}, t_n = i)$

(the probability of generating $w_{1...n}$ and being in state i)

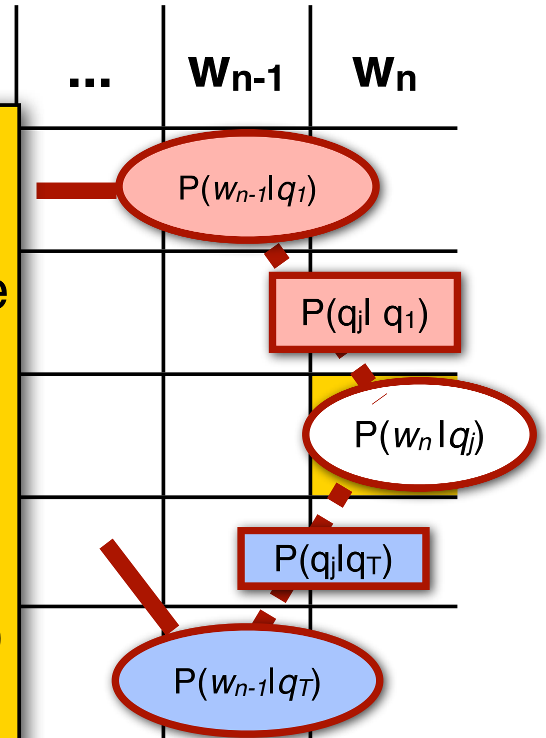
3. Finding P(w) = ∑t P(t,w)

$$\begin{aligned}
\underbrace{P(\mathbf{w_{1..n}}, \mathbf{t_n} = \mathbf{t_i})}_{\texttt{trellis[n][i]}} &= \sum_{\mathbf{t_{1..n}}|\mathbf{t_n}=\mathbf{t_i}} P(\mathbf{w_{1...n}}, \mathbf{t_{1..n}}) \\
&= P(w_n|t_i)\sum_{j=1}^{T} P(t_i|t_j)\,\underbrace{P(\mathbf{w_{1..n-1}}, t_{n-1} = t_j)}_{\texttt{trellis[n-1][j]}}
\end{aligned}$$



The Viterbi algorithm

Finding $p^* = \max_t P(t,w)$



Now, the trellis stores the **maximal probability of any path ending in t_i**, not the total probability mass of all paths ending in t_i

Let

$$MaxTrellis[n][i] := \max(P(w_{1..n}, t_n = i))$$

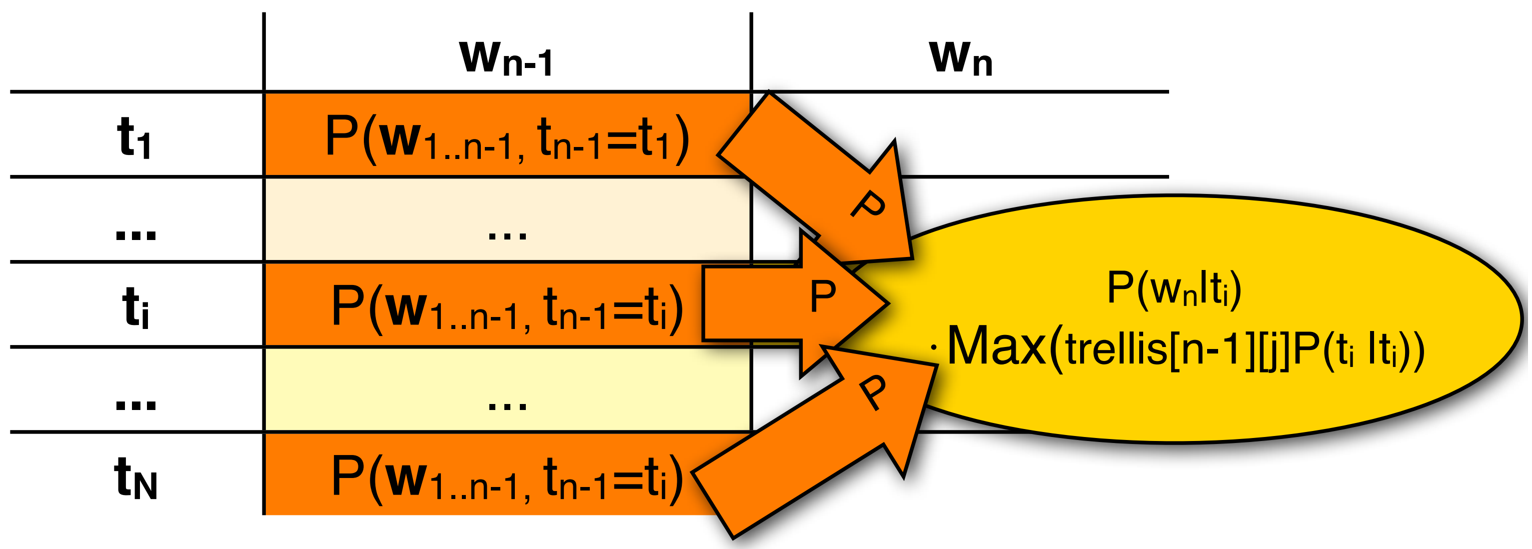
(the highest probability of any tag sequence that generates $w_{1...n}$ and ends in state i)

Finding p* = max$_t$ P(t,w)

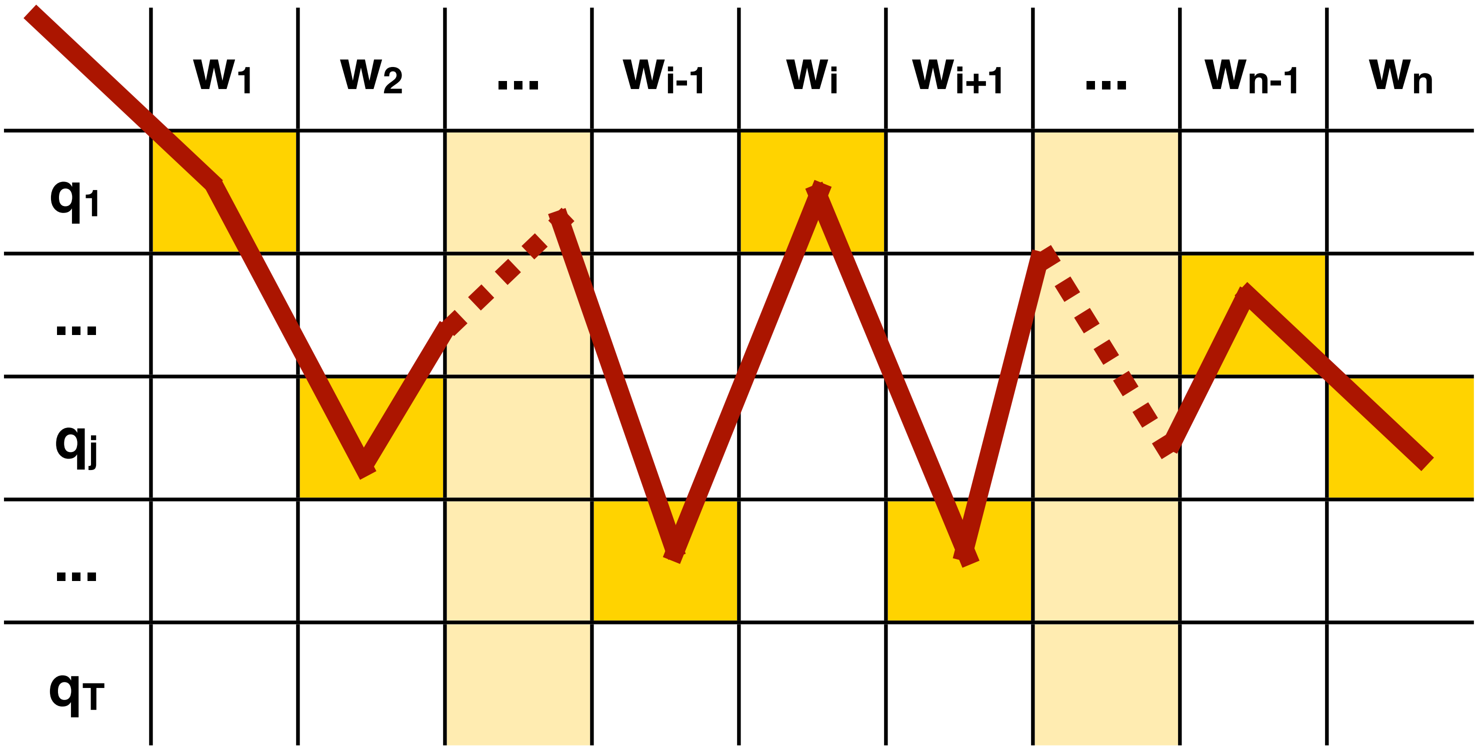
$$\max(\underbrace{P(\mathbf{w_{1..n}}, \mathbf{t_n} = \mathbf{t_i})}_{\texttt{maxTrellis[n][i]}}) = \sum_{\mathbf{t_{1..n}} | \mathbf{t_n} = \mathbf{t_i}} P(\mathbf{w_{1...n}}, \mathbf{t_{1..n}})$$

$$= P(w_n | t_i) \max_{j=1} \left(P(t_i | t_j) \underbrace{P(\mathbf{w_{1..n-1}}, t_{n-1} = t_j)}_{\texttt{maxTrellis[n-1][j]}} \right)$$

$$= b_{in} \max_{j=1} \left(a_{ji} \underbrace{P(\mathbf{w_{1..n-1}}, t_{n-1} = t_j)}_{\texttt{maxTrellis[n-1][j]}} \right)$$



Retrieving $t^* = \arg\max_t P(t,w)$



- By keeping only **one backpointer** from each cell to the tag in the previous column that yields the highest probability, we can retrieve the most likely tag sequence when we're done.

The Forward-Backward algorithm

Learning an HMM from unlabeled data

```
Pierre Vinken , 61 years old , will
join the board as a nonexecutive
director Nov. 29 .
```

Tagset:
NNP: proper noun
CD: numeral,
JJ: adjective,...

We can't count anymore.

We have to *guess* how often we'd *expect* to see $t_i t_j$ *etc.* in our data set. Call this **expected count** $\langle C(...)\rangle$

- Our estimate for the transition probabilities:

$$\hat{P}(t_j|t_i) \quad = \quad \frac{\langle C(t_i t_j)\rangle}{\langle C(t_i)\rangle}$$

- Our estimate for the emission probabilities:

$$\hat{P}(w_j|t_i) \quad = \quad \frac{\langle C(w_j\_t_i)\rangle}{\langle C(t_i)\rangle}$$

Learning an HMM: the EM algorithm

Initialization:

- Take a data set **S**
- Guess some initial A_0 and B_0
  Let $\lambda_i = \lambda_0 = (A_0 , B_0)$

The Expectation (E) step:

- Use λ_i to compute $\langle C(t) \mid \lambda_i, \mathbf{S} \rangle$

The Maximization (M) step:

- Calculate a new HMM λ_{i+1} using $\langle C(t) \mid \lambda_i, \mathbf{S} \rangle$

Repeat the E and M steps until λ converges

How do we compute $\langle C(t_i) \rangle$?

- Our corpus ***S*** consists of *K* sentences:

S = { S_1: “Pierre Vinken…”
S_2: “Vinken joined the board…”
…. …..
S_K: “Yesterday, the Dow Jones...”}

- We have to sum how often we expect t_i in each sentence

$$\langle C(t_i)|\mathbf{S}\rangle_P = \sum_k^K \langle C(t_i|S_k)\rangle_P$$

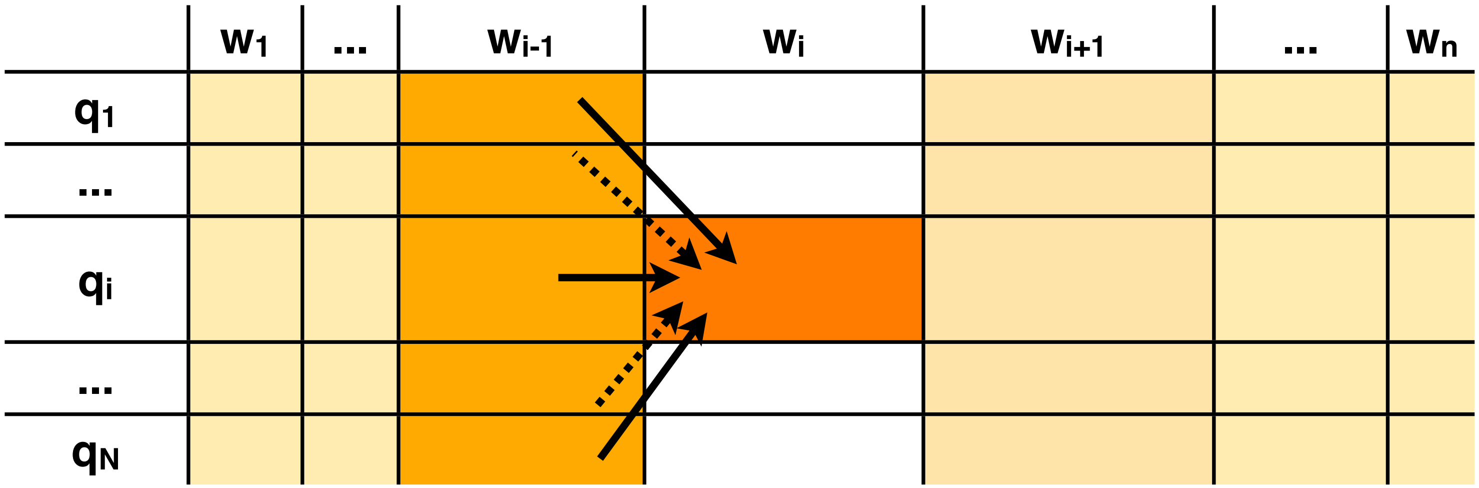
How do we compute $\langle C(t_i) \mid S_k \rangle$

	w_1	...	w_{i-1}	w_i	w_{i+1}	...	w_n
q_1							
...							
q_i							
...							
q_N							

- t_i can be assigned to any word in the sentence (it corresponds to one row in the trellis)
- We have to sum how often we expect t_i in each cell of this row

$$\langle C(t_i) | \mathbf{w}_{1..n} \rangle_P = \sum_{j}^{n} \langle C(t_i | w_j) \rangle_P$$

How do we compute $\langle C(t_i) \mid w_j \rangle$



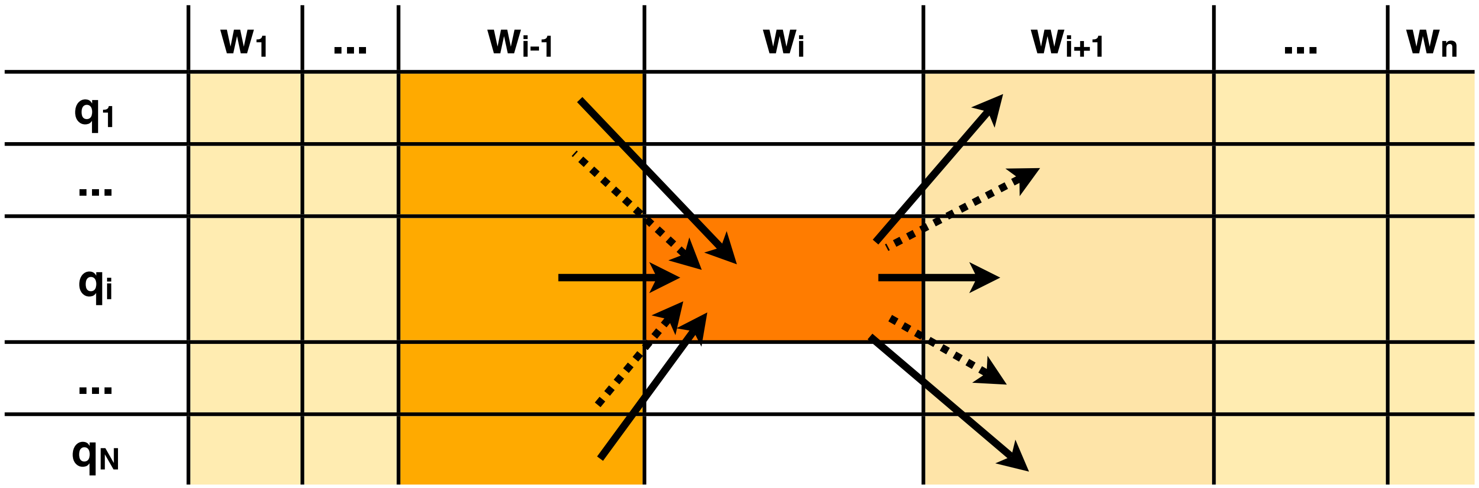
- We need to know $P(\mathbf{t}_j = t_i | \mathbf{w}_{1..n})$

- We can use Bayes Rule:

$$P(\mathbf{t}_j = t_i | \mathbf{w}_{1...n}) = \frac{P(\mathbf{t_j} = t_i, \mathbf{w}_{1...n})}{P(\mathbf{w}_{1...n})}$$

- The forward trellis tells us $\underbrace{P(\mathbf{w_{1..j}}, \mathbf{t_j} = t_i)}_{\texttt{trellis[j][i]}}$ and $P(\mathbf{w}_{1..n})$

How do we compute $\langle C(t_i) | w_j \rangle$



- We need to know $P(\mathbf{w_{1..n}}, \mathbf{t_j} = t_i)$
- The trellis tells us $\underbrace{P(\mathbf{w_{1..j}}, \mathbf{t_j} = t_i)}_{\texttt{trellis[j][i]}}$

- We can use the Chain rule:

$$P(\mathbf{w_{1..j}}\ \mathbf{_{j+1..n}}, \mathbf{t_j} = t_i) = P(\mathbf{w_{1..j}}, \mathbf{t_j} = t_i)P(\mathbf{w_{j+1...n}} | \mathbf{w_{1..j}}, \mathbf{t_j} = \mathbf{t_i})$$

Computing $P(\mathbf{w}_{j+1...n} | \mathbf{w}_{1..j}, \mathbf{t}_j = t_i)$

In our HMM model, words depend only on their tags, thus:

$$P(\mathbf{w_{j+1\ldots n}}|\mathbf{w_{1..j}}, \mathbf{t_j} = \mathbf{t_i}) \quad = \quad P(\mathbf{w_{j+1\ldots n}}|\mathbf{t_j} = t_i)$$

We can calculate this recursively:

$$P(\mathbf{w_{j+1\ldots n}}|\mathbf{t_j} = t_i) \quad = \quad \sum_k P(t_k|t_i)P(\mathbf{w_{j+1}}|t_k)P(\mathbf{w_{j+2\ldots n}}|\mathbf{t_{j+1}} = t_k)$$

Putting it all together

1. In our model, P(**w** | $\mathbf{t_j}$ = t_i) decomposes into two terms:
a **forward** and a **backward** probability

$$P(\mathbf{w_{1\ldots j\ \ j+1\ldots n}} | \mathbf{t_j} = t_i)$$

$$= \underbrace{P(\mathbf{w_{1\ldots j}} | \mathbf{t_j} = t_i)}_{\text{Forward probability of } \mathbf{w}_{1..j}, t_i} \times \underbrace{P(\mathbf{w_{j+1\ldots n}} | \mathbf{t_j} = t_i)}_{\text{Backward probability of } \mathbf{w}_{j..n}, t_i}$$

Forward and backward probabilities

2. Both can be calculated recursively:

$$\underbrace{P(\mathbf{w_{1\ldots j}}|\mathbf{t_j} = t_i)}_{\text{Forward probability of } \mathbf{w}_{1..j}, t_i}$$

$$= \quad \sum_k P(t_i|t_k)P(\mathbf{w_j}|t_k) \quad \underbrace{P(\mathbf{w_{1\ldots j-1}}|\mathbf{t_{j-1}} = t_k)}_{\text{Forward probability of } \mathbf{w}_{1..j-1}, t_k}$$

$$\underbrace{P(\mathbf{w_{j+1\ldots n}}|\mathbf{t_j} = t_i)}_{\text{Backward probability of } \mathbf{w}_{j..n}, t_i}$$

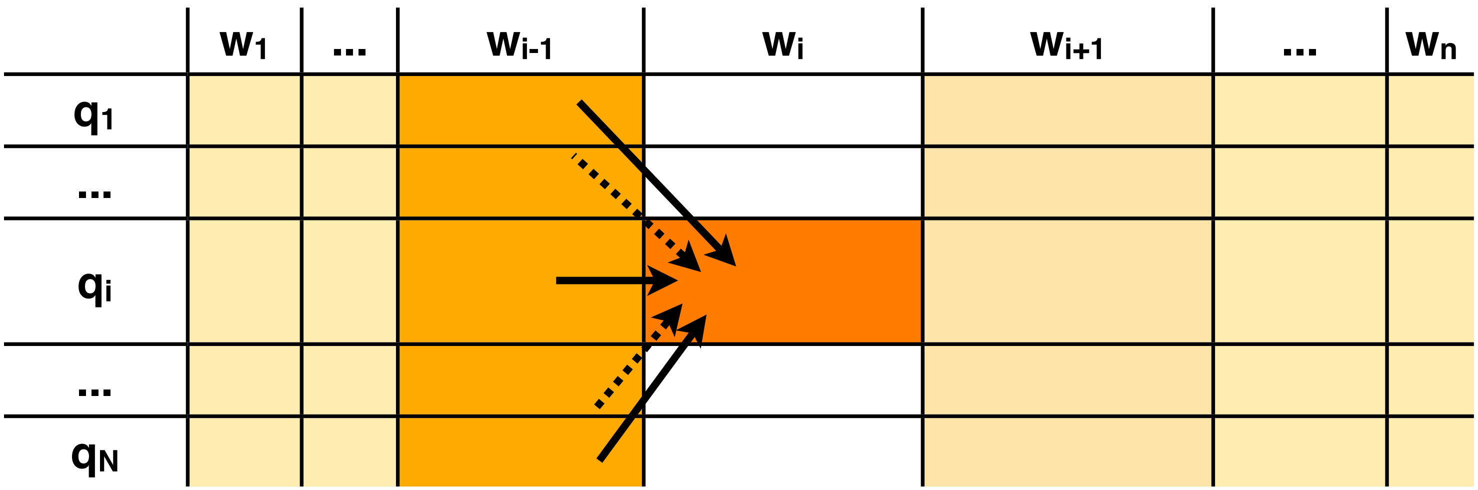
$$= \quad \sum_k P(t_k|t_i)P(\mathbf{w_{j+1}}|t_k) \quad \underbrace{P(\mathbf{w_{j+2\ldots n}}|\mathbf{t_{j+1}} = t_k)}_{\text{Backward probability of } \mathbf{w}_{j+1..n}, t_k}$$

Using the trellis

3. The trellis tells us already the forward probabilities:

$$\underbrace{P(\mathbf{w_{1\ldots j}}|\mathbf{t_j}=t_i)}_{\texttt{FWtrellis[j][i]}}$$

$$= \quad \sum_k a_{ki} b_{kj} \underbrace{P(\mathbf{w_{1\ldots j-1}}|\mathbf{t_{j-1}}=t_k)}_{\texttt{FWtrellis[j-1][k]}}$$

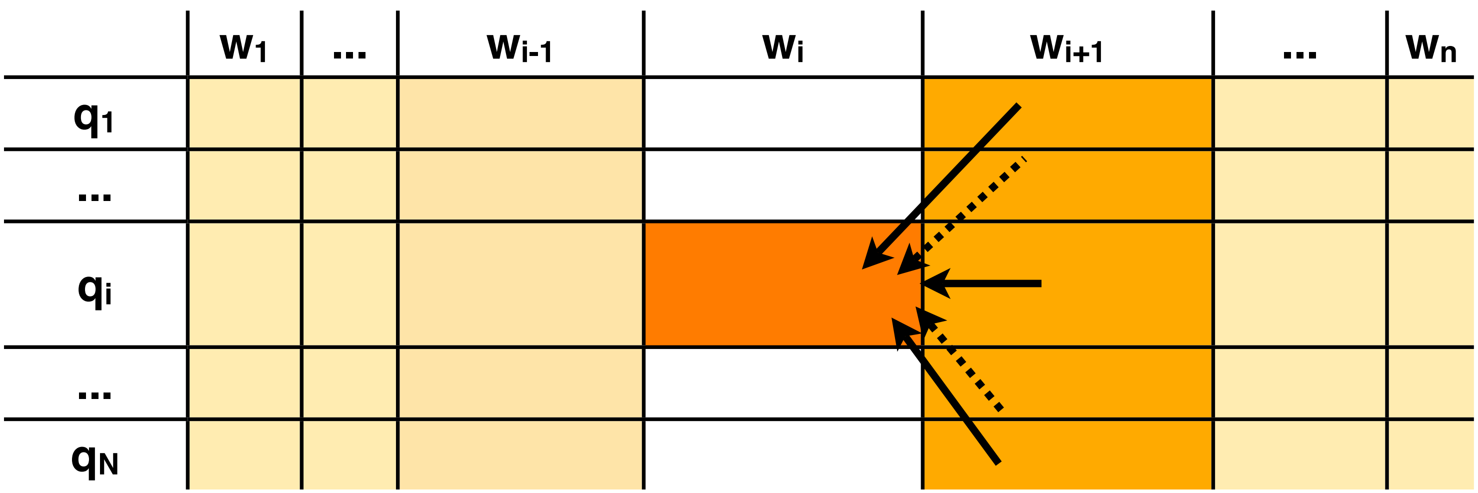


Using the trellis

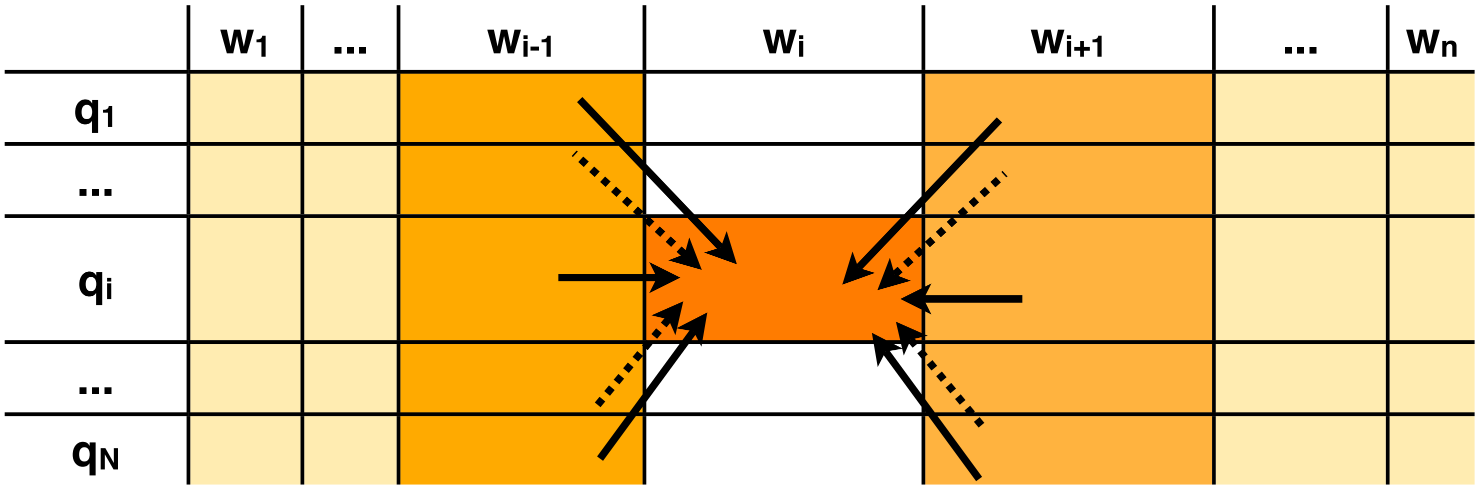
4. We can also use it to keep track of the backward probabilities:

$$\underbrace{P(\mathbf{w_{j+1\ldots n}}|\mathbf{t_j} = t_i)}_{\texttt{BWtrellis[j][i]}}$$

$$= \quad \sum_k a_{ik} b_{kj+1} \underbrace{P(\mathbf{w_{j+2\ldots n}}|\mathbf{t_{j+1}} = t_k)}_{\texttt{BWtrellis[j+1][k]}}$$



How do we compute $\langle C(t_i) | w_j \rangle$



- The trellis tellls us everything we need to know to compute

$$P(\mathbf{t}_j = t_i | \mathbf{w}_{1\ldots n}) = \frac{P(\mathbf{t_j} = t_i, \mathbf{w}_{1\ldots n})}{P(\mathbf{w}_{1\ldots n})}$$