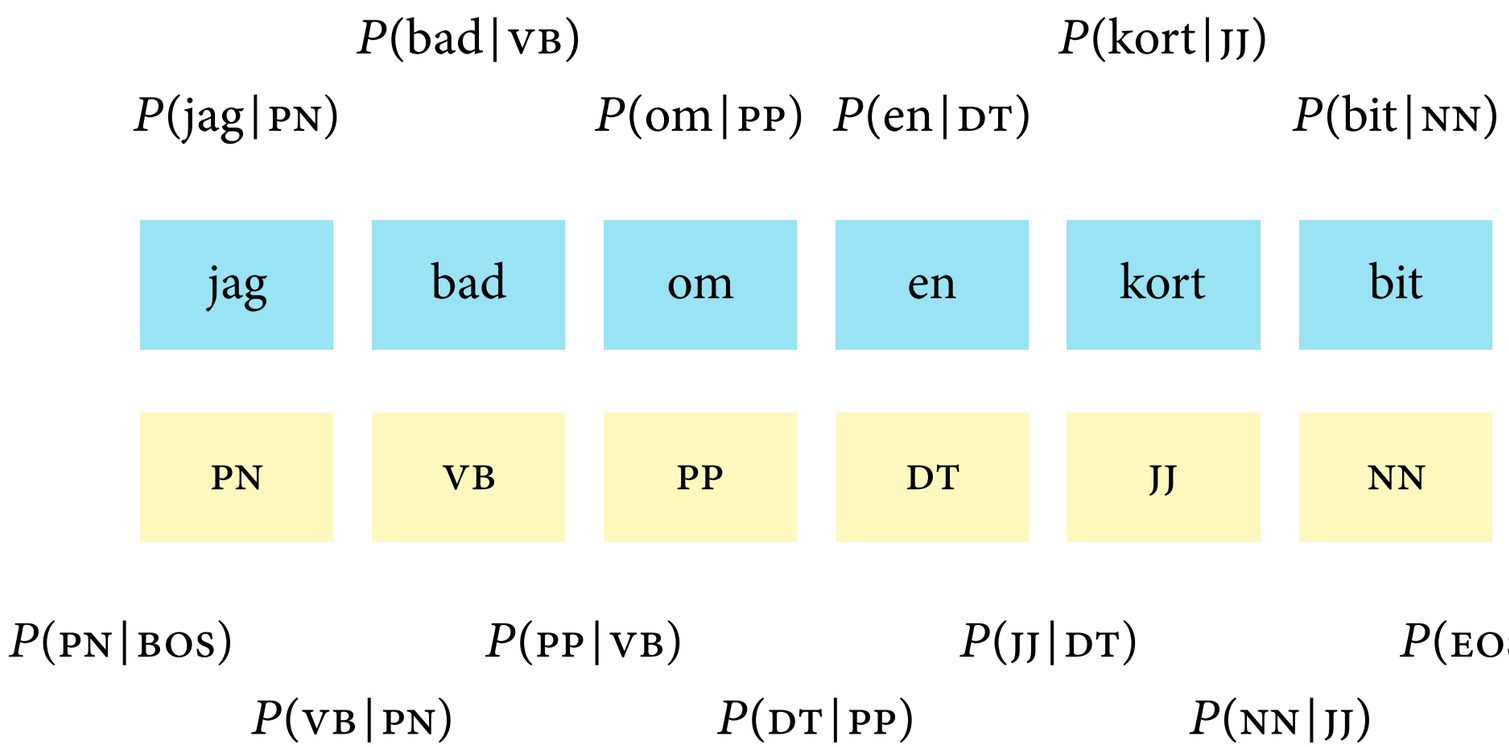
The Viterbi Algorithm

Probability of a tagged sentence



product of transition and output probabilities

Tagging with a hidden Markov model

- Given a sentence, we want to find a sequence of tags such that the probability of the tagged sentence is maximal.

  The tag sequence is not given in advance; it is 'hidden'!

- For each sentence there are many different tag sequences with many different probabilities.

  combinatorial explosion

- In spite of this, the most probable tag sequence can be found efficiently using the **Viterbi algorithm**.

High-level description

- The algorithm takes as its inputs a HMM and a sentence and computes the most probable tag sequence for the sentence.
- The algorithm fills a matrix that contains one row for each possible tag and one column for each position in the sentence.

  including BOS, EOS

- In this presentation we fill the matrix with negative log probabilities; we can interpret them as costs in crowns.

  We do this to avoid underflow.

The central invariant

- The algorithm should make sure that the value in row t, column i is the minimal cost needed to tag the first i words in the sentence in such a way that word number i is tagged as t.

  Remember that minimal cost = maximal probability.

- If the algorithm can achieve this, then we can read off the least possible cost to tag the complete sentence from the last column.

		jag$_1$	bad$_2$	om$_3$	en$_4$	kort$_5$	bit$_6$	
BOS	0,00							
DT		14,49	21,33	29,38	24,82	42,62	50,67	
JJ		15,46	21,13	29,88	35,22	33,00	48,36	
NN		11,22	19,53	29,74	33,58	35,44	41,63	
PN		5,35	21,43	28,86	29,86	42,50	50,81	
PP		14,59	20,02	20,70	38,53	42,41	48,32	
VB		16,11	14,83	29,53	39,65	43,08	49,15	
EOS								45,93

Hidden Markov model 1: Transition costs

	PN	VB	PP	DT	JJ	NN	EOS
BOS	1,69	3,58	2,25	2,50	3,37	1,76	11,19
PN	4,00	0,69	2,34	4,00	3,69	3,85	7,94
VB	1,95	2,17	2,04	2,56	2,97	2,18	6,87
PP	3,09	6,42	5,49	1,82	2,43	0,85	8,38
DT	5,61	10,22	5,26	5,82	0,93	0,84	10,22
JJ	5,73	3,62	2,98	5,68	3,28	0,43	6,35
NN	5,30	1,70	1,49	5,17	4,23	3,11	4,30

Hidden Markov model 2: Observation costs

	jag	bad	om	en	kort	bit
PN	3,66	12,08	12,08	6,08	12,08	12,08
VB	12,53	8,79	12,53	12,53	12,53	12,53
PP	12,33	12,33	3,83	12,33	12,33	12,33
DT	11,99	11,99	11,99	2,29	11,99	11,99
JJ	12,09	12,09	12,09	12,09	7,25	12,09
NN	9,47	10,33	12,73	12,03	9,78	8,19

		jag$_1$	bad$_2$	om$_3$	en$_4$	kort$_5$	bit$_6$
BOS	0,00						
DT		14,49					
JJ							
NN							
PN							
PP							
VB							
EOS							

$$0.00 + P(\text{DT}|\text{BOS}) + P(\text{jag}|\text{DT}) = 2.50 + 11.99 = 14.49$$

		jag$_1$	bad$_2$	om$_3$	en$_4$	kort$_5$	bit$_6$
BOS	0,00						
DT		14,49					
JJ		15,46					
NN		11,22					
PN		5,35					
PP							
VB							
EOS							

$$0.00 + P(\text{PN}|\text{BOS}) + P(\text{jag}|\text{PN}) = 1.69 + 3.66 = 5.35$$

		jag$_1$	bad$_2$	om$_3$	en$_4$	kort$_5$	bit$_6$
BOS	0,00						
DT		14,49	21,33	29,38	35,15		
JJ		15,46	21,13	29,88			
NN		11,22	19,53	29,74			
PN		5,35	21,43	28,86			
PP		14,59	20,02	20,70			
VB		16,11	14,83	29,53			
EOS							

$$28.86 + P(\text{DT}|\text{PN}) + P(\text{en}|\text{DT}) = 28.86 + 4.00 + 2.29 = 35.15$$

		jag$_1$	bad$_2$	om$_3$	en$_4$	kort$_5$	bit$_6$
BOS	0,00						
DT		14,49	21,33	29,38	24,82		
JJ		15,46	21,13	29,88			
NN		11,22	19,53	29,74			
PN		5,35	21,43	28,86			
PP		14,59	20,02	20,70			
VB		16,11	14,83	29,53			
EOS							

$$20.70 + P(\text{DT}|\text{PP}) + P(\text{en}|\text{DT}) = 20.70 + 1.82 + 2.29 = 24.82$$

		jag_1	bad_2	om_3	en_4	$kort_5$	bit_6	
BOS	0,00							
DT		14,49	21,33	29,38	24,82	42,62	50,67	
JJ		15,46	21,13	29,88	35,22	33,00	48,36	
NN		11,22	19,53	29,74	33,58	35,44	41,63	
PN		5,35	21,43	28,86	29,86	42,50	50,81	
PP		14,59	20,02	20,70	38,53	42,41	48,32	
VB		16,11	14,83	29,53	39,65	43,08	49,15	
EOS								45,93

$$41.63 + P(\text{EOS}|\text{NN}) = 41.63 + 4.30 = 45.93$$

		jag$_1$	bad$_2$	om$_3$	en$_4$	kort$_5$	bit$_6$
BOS	0,00						
DT		14,49	21,33	29,38	24,82	42,62	50,67
JJ		15,46	21,13	29,88	35,22	33,00	48,36
NN		11,22	19,53	29,74	33,58	35,44	41,63
PN		5,35	21,43	28,86	29,86	42,50	50,81
PP		14,59	20,02	20,70	38,53	42,41	48,32
VB		16,11	14,83	29,53	39,65	43,08	49,15
EOS							45,93

Follow the backpointers to read off the sequence.

		jag$_1$	skrev$_2$	på$_3$	utan$_4$	att$_5$	tveka$_6$
BOS	0,00						
IE		17,22	21,69	30,02	33,79	34,63	54,70
PL		21,77	21,20	22,10	39,77	49,28	55,06
PN		5,35	21,43	27,87	33,85	44,12	48,09
PP		14,59	20,02	18,69	28,95	44,66	50,70
SN		15,83	21,51	29,20	34,29	35,24	51,40
VB		16,11	13,84	28,54	37,64	43,96	44,86
EOS							51,74

It does not suffice to pick the best cell in each column!

Computational complexity

- Let m, n denote the number of tags in the HMM and the length of the input sentence, respectively.

- The memory required by the Viterbi algorithm is in $O(mn)$; this corresponds to the size of the matrix.

- The runtime required by the Viterbi algorithm is in $O(m^2n)$: We need to fill $O(mn)$ cells, and each cell requires us to look at $O(m)$ cells in the previous column.