Answer 2 –

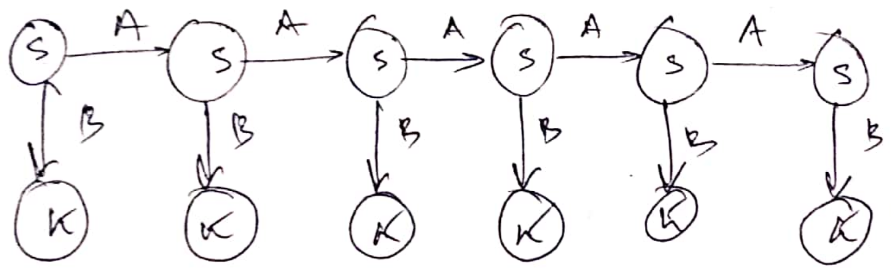
Basic idea – $\boxed{P(t|w) = \dfrac{P(w|t) \times P(t)}{P(w)}}$

$$P(w_1 w_2 \ldots w_n) = P(w_1) \prod_{i=2}^{n} P(w_i | w_1 \ldots w_{i-1})$$

For 2 assumption length →

$$P(w_1, w_2, \ldots, w_n) \approx P(w_1) \cdot P(w_2|w_1) \prod_{i=2}^{n-1} P(w_{i+1}|w_i, w_{i-1})$$

$$P(w_i | w_{i-2}, w_{i-1}) = \frac{\text{Count}(w_{i-2}, w_{i-1}, w_i)}{\text{Count}(w_{i-2}, w_{i-1})}$$



Transition probability →

For every transition $S_j \to S_k$, transition probability $P(s_k|s_i)$ is present.

Emission Probability →

For every state S_k and every word w_i, we have an emission probability $P(w_i|s_k)$ (implementing $P(w_i|t_i)$)

The probabilities are calculated using log probabilities and optimized data structures such as dictionary are used for every tag set.

The columns contain both w_{i-1}, w_{i-2} for w_{i-1} calculation and hence shows trigram model through dependencies on previous two words.

Start and stop symbols - '*' and 'STOP' are added and then the starting probabilities are calculated, the transition_prior_probabilities are also taken into account.

Also →

$$p(x_1, x_2, \ldots, x_n, y_1, y_2 \ldots y_{n+1})$$
$$= \prod_{i=1}^{n+1} q(y_i \mid y_{i-2}, y_{i-1}) \cdot \prod_{i=1}^{n} e(x_i \mid y_i)$$

K = set of possible tags. $K_{-1} = K_0 = \{*\}$ and $K_k = K$ for $k = 1, 2, \ldots, n$

Initialization → $\pi(0, *, *) = 1$

Algorithm →

- For $k = 1, \ldots, n$
  - For $u \in K_{k-1}, v \in K_k$,

$$\pi(k, u, v) = \max_{w \in K_{k-2}} \left(\pi(k-1, w, u) \times q(v \mid w, u) \times e(x_k \mid v)\right)$$

- Return $\max_{u \in K_{n-1}, v \in K_n} \left(\pi(n, u, v) \times q(STOP \mid u, v)\right)$

Basic Viterbi