

Homework #3: CMPT-825

Anoop Sarkar – anoop@cs.sfu.ca

(1) **Part-of-speech Tagging:**

Consider the task of assigning the most likely part of speech tag to each word in an input sentence. We want to get the best (or most likely) tag sequence as defined by the equation:

$$T^* = \operatorname{argmax}_{t_0, \dots, t_n} P(t_0, \dots, t_n \mid w_0, \dots, w_n)$$

- Write down the equation for computing the probability $P(t_0, \dots, t_n \mid w_0, \dots, w_n)$ using Bayes Rule and a trigram probability model over part of speech tags.
- We realize that we can get better tagging accuracy if we can condition the current tag on the previous tag and the next tag, i.e. if we can use $P(t_i \mid t_{i-1}, t_{i+1})$. Thus, we define the best (or most likely) tag sequence as follows:

$$\begin{aligned} T^* &= \operatorname{argmax}_{t_0, \dots, t_n} P(t_0, \dots, t_n \mid w_0, \dots, w_n) \\ &\approx \operatorname{argmax}_{t_0, \dots, t_n} \prod_{i=0}^n P(w_i \mid t_i) \times P(t_i \mid t_{i-1}, t_{i+1}) \text{ where } t_{-1} = t_{n+1} = \text{none} \end{aligned}$$

Explain why the Viterbi algorithm cannot be directly used to find T^* for the above equation.

- BestScore is the score for the maximum probability tag sequence for a given input word sequence.

$$\text{BestScore} = \max_{t_0, \dots, t_n} P(t_0, \dots, t_n \mid w_0, \dots, w_n)$$

It is a bit simpler to compute than Viterbi since it does not compute the best sequence of tags (no back pointer is required). For the standard trigram model $P(t_i \mid t_{i-2}, t_{i-1})$:

$$\text{BestScore} = \max_{t_0, \dots, t_n} \prod_{i=0}^n P(w_i \mid t_i) \times P(t_i \mid t_{i-2}, t_{i-1})$$

Assuming that $t_{-1} = t_{-2} = t_{n+1} = \text{none}$, we can compute BestScore recursively from left to right as follows:

$$\begin{aligned} \text{BestScore}[i+1, t_{i+1}, t_i] &= \max_{t_{i-1}, t_i} (\text{BestScore}[i, t_{i-1}, t_i] \times P(w_{i+1} \mid t_{i+1}) \times P(t_{i+1} \mid t_{i-1}, t_i)) \\ &\text{for all } -1 \leq i \leq n \\ \text{BestScore} &= \max_{\langle t, \text{none} \rangle} \text{BestScore}[n+1, \langle t, \text{none} \rangle] \end{aligned}$$

This algorithm for computing BestScore is simply the recursive forward algorithm for HMMs but with the sum replaced by max.

Provide an algorithm in order to compute BestScore for the improved trigram model $P(t_i \mid t_{i-1}, t_{i+1})$:

$$\text{BestScore} = \max_{t_0, \dots, t_{n-1}} \prod_{i=0}^{n+1} P(w_i \mid t_i) \times P(t_i \mid t_{i-1}, t_{i+1})$$

where $t_{-1} = w_{n+1} = t_{n+2} = t_{n+1} = t_n = \text{none}$, and assume that: $P(w_{n+1} = \text{none} \mid t_{n+1} = \text{none}) = 1$; $P(t_0 \mid t_{-1} = \text{none}, t_1) \approx P(t_0 \mid t_1)$; $P(t_{n+1} \mid t_n, t_{n+2}) = 1$ and $P(t_n \mid t_{n-1}, t_{n+1} = \text{none}) \approx P(t_n \mid t_{n-1})$

You can provide either pseudo code, a recursive definition of the algorithm, or a recurrence relation.

Hint: The first step would be to extend the recursive BestScore algorithm given above to read the input from right to left.

- d. Provide an algorithm in order to compute BestScore for the even more improved model:

$$\text{BestScore} = \max_{t_0, \dots, t_{n-1}} \prod_{i=0}^{n+1} P(w_i \mid t_{i-1}, t_i, t_{i+1}) \times P(t_i \mid t_{i-2}, t_{i-1}, t_{i+1})$$