Lab4 Report

Implementation

In this lab, the code is based on the last lab and the only change is the part of prediction method. Viterbi algorithm and Beam Search algorithm has been implemented and used to replace the algorithm in prediction method last lab.

For Viterbi algorithm, in this case, only phi1(current word and current label) need to be considered. The function (Phi) which is following and multiplying the W could be ignored that means only the weight of the current word and the current label need to be considered, because in the case of phi1, every combination of current word and current label is independent in the sentence and the function is always return 1. The formulation, after simplifying, it would be the formulation in following.

$$V[y,n] = \max_{y' \in \mathcal{Y}} V[y', n-1] + \mathbf{w}$$

Only dictionaries and 2-D arrays are adopted and no matrix is used in the program.

For Beam Search algorithm, it has the same considerations as Viterbi. However, different size of beam has been selected in the program (size = 1, 5, 10). The accuracy and speed would be shown in the evaluation part.

Evaluation

Both Viterbi algorithm and Beam Search algorithm have better performance in speed than the algorithm than last lab. Both two algorithms have the same result and it is the same as last lab in case of phi1. However, the better speed up performance in Beam Search algorithm only exists in the situation which is in small beam size. Theoretically, comparing the complexity of three algorithms, we can get the performance in speed of each algorithms. Given m to represent the number of label, *n* to represent the average length of the sentence and k to represent the size of beam. The complexity of prediction in last lab can be computed as m^n; the complexity of Viterbi algorithm is $n * m^2$; the complexity of Beam Search algorithm is $\mathrm{n} * \mathrm{k} * (\mathrm{m} + O_{(s)})$, $O_{(s)}$ represent the complexity of sort algorithm. From the complexity of algorithms above, it can be concluded that Viterbi algorithm and Beam Search algorithm are better than the algorithm of the last lab, meanwhile, Beam Search algorithm is better than Viterbi algorithm in the situation which is that the k is much smaller than *m* and choosing a good sort approach. In this program, the build-in sort function, *sort()*, would be used in Beam Search algorithm, and it complexity is m^2. Thus, when the beam size comes to over 5, it would cost much more time than Viterbi algorithm. To be more specific, the comparing results would be displayed in the table bellowed.

Epoch	**Last Lab**		**Viterbi**		**Beam Search**					
					Size = 1		**Size = 5**		**Size = 10**	
	F1 score	**Time**	**F1 score**	**Time**	**F1 score**	**Time**	**F1 score**	**Time**	**F1 score**	**Time**
5	0.644721	176.35s	0.644721	4.18s	0.644721	1.36s	0.644721	6.93s	0.644721	14.75s
10	0.648402	347.87s	0.648402	8.46s	0.648402	3.00s	0.648402	13.96s	0.648402	29.42s
15	0.645340	532.32s	0.645340	12.90s	0.645340	4.18s	0.645340	20.18s	0.645340	43.68s