UNIVERSITY OF TEXAS AT AUSTIN

Natural Language Processing Assignment 3 (P3)

Michail Mersinias

October 6, 2021

QUESTION 1

Report accuracy and runtime for your CRF model using both Viterbi and at least 4 different values of the beam size, including beam size 1. (Plot these values as a graph or in a table.)

Answer: The results of the CRF model are presented in Table 1 below. An exhaustive search was performed for all possible sizes of the Beam data structure. As the number of tags is 9, when the beam size equals 9, Beam Search decoding is equivalent to Viterbi decoding. For any beam size greater than 9, the Beam Search decoding is equivalent to that where beam size equals 9, because the rest of the elements are computationally redundant.

Beam Size	Time	F1 Score	Precision	Recall
1	16.29	76.67	77.74	75.64
2	17.39	84.12	84.84	83.41
3	18.08	86.52	87.32	85.73
4	18.99	87.69	88.61	86.79
5	19.78	88.05	89.00	87.13
6	20.71	88.05	89.00	87.13
7	21.56	88.05	89.00	87.13
8	22.31	88.05	89.00	87.13
9	22.99	88.05	89.00	87.13
Viterbi	22.67	88.05	89.00	87.13

Table 1: CRF - Beam Search vs Viterbi Decoding Results

QUESTION 2

Describe what trends you see in accuracy and runtime. How does beam search compare to your expectations here?

Answer:

Regarding runtime, as shown in Figure 1, Beam Search decoding runtime increases in a linear fashion based on the beam size parameter. In Table 1, we observed that precision and recall were relatively balanced, thus we can look straight into F1 score as our metric of accuracy. Regarding F1 score, as shown in Figure 2, Beam Search decoding F1 score increases in an exponential fashion until the beam size parameter equals 4. After that, there is a small linear increase and the maximum value is reached when the beam size equals 5.

Combining our two observations, we can conclude that it is certainly worth increasing the beam size parameter up until 4, because our gain in F1 score is exponential while our cost in runtime is linear. This allows us to be very close to the maximum (as in Viterbi) F1 score while having an appoximately 19.4% better runtime performance. We could potentially increase the beam size to 5 to actually reach the same F1 score as Viterbi and still have a runtime of less than 20 seconds, thus maintaining an approximately 14.6% better runtime performance.

Finally, my initial expectation was that Beam Search is a greedy heuristic search algorithm and thus does not guarantee an optimal solution. However, I was pleasantly surprised to see that with 1/3 of the states (beam size = 3), it can provide a great approximation with 86.52% F1 Score, almost within a unit difference. And with half of the states (beam size = 5), it reaches the optimal solution with 88.05% F1 Score. Regarding time, the linear increase behavior was to be expected as the algorithm implementation maintains the same complexity and beam size is simply a parameter for an existing iteration structure.

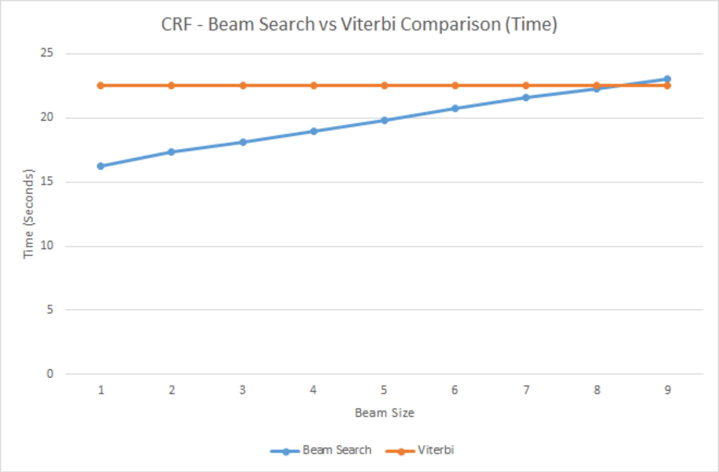

Figure 1: CRF - Beam Search vs Viterbi Decoding (Time Comparison)

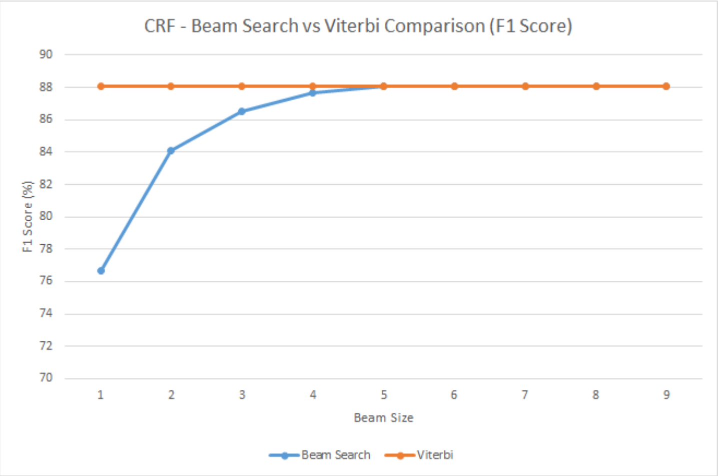

Figure 2: CRF - Beam Search vs Viterbi Decoding (F1 Score Comparison)

Question 3

Under what circumstances do you think beam search would be more effective relative to Viterbi? That is, what characteristics of a problem (sequence length, number of states, model, features, etc.) would change what you've observed?

Answer:

The difference between Viterbi based decoding and Beam Search based decoding is that the first one performs an exhaustive search over all previous states while the latter one performs a search only on the top scoring N (N = beam size) previous states. The score in position i is based on the sum of the score in position $i-1$, the transition score and the emission score. For the sake of comparison, we can ignore the emission score as it is an addition that will be performed in both cases equivalently. Thus, the score in position i is affected by the sum of the score in position $i-1$ plus the transition score.

We denote the n^{th} best score in position i as $V_{i,n}$ and the transition score from the state that corresponds to $V_{i-1,n}$ towards i as $T_{i-1,n,i}$. In order for Viterbi to be "better" than Beam Search of beam size equal to N, the k^{th} ($k > N$) element of the Beam in position $i-1$ ($V_{i-1,k}$) plus its transition score $T_{i-i,k,i}$ should be the one which is part of the optimal path. Thus:

$$V_{i-1,k} + T_{i-i,k,i} > V_{i-1,t} + T_{i-i,t,i}, \forall t \in [0, N) \tag{1}$$

Thus, in order for Beam Search to be more effective, it is sufficient for us to find one t which makes Equation 1 false. Therefore, we can rewrite it as follows:

$$t \in [0, N) : V_{i-1,k} + T_{i-i,k,i} < V_{i-1,t} + T_{i-i,t,i} \tag{2}$$

Now we want to find a t which makes Equation 2 true. However, we already know that $V_{i-1,k} < V_{i-1,t}$ as $t < k$ because $t \in [0, N)$ and $k > N$. We denote the result of the subtraction $V_{i-1,t} - V_{i-1,k}$ as D_v which is a positive number ($D_v > 0$). This allows us to simplify the equation further as follows:

$$t \in [0, N) : T_{i-i,t,i} + D_v > T_{i-i,k,i} \tag{3}$$

There are two ways for Equation 3 to be true. Firstly, if $T_{i-i,k,i}$ is not much larger than $T_{i-i,t,i}$. In order to avoid this risk, a reduction in the number of states would decrease the probability that there happen to be two unfavorably vastly different state transition scores at the given time, that is a difference greater than D_v. Secondly, if D_v is large enough to cover up for any major unfavorable difference in state transition scores, which happens more as the beam size parameter increases. Finally, a larger sequence length would mean a greater number of transitions, thus a greater risk of falling into a case of large unfavorable difference is state transition scores. That is why, to avoid this risk, a reduction in sequence length is helpful.

Thus, a reduction in both the number of states and the sequence length, as well as an increase in beam size, would be beneficial to the effectiveness of the Beam Search decoding.