Haley Ramirez

10/17/2021

A3 CS388

Viterbi vs Beam Decode of CRF for NER

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. (You can plot these values as a graph or in a table.)

|  |  |  |
| --- | --- | --- |
|  | **Accuracy (F1)** | **Runtime (s)** |
| **Viterbi** | 88.1 | 24.5 |
| **Beam = 1** | 76.8 | 11.5 |
| **Beam = 2** | 84.2 | 17.5 |
| **Beam = 3** | 86.8 | 20.6 |
| **Beam = 4** | 87.7 | 26.4 |
| **Beam = 5** | 88.1 | 30 |

Table 1. The accuracy and runtimes of the different decoder methods.

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

The Viterbi Algorithm achieves a higher accuracy when compared to smaller beam sizes (beam size <=4), but once the beam size is 5 or more the beam search algorithm matches Viterbi. This means that the smaller beam sizes do fail to keep some of the mostly likely combinations of tags around, and the F1 reflects this. The larger beam size of 5 meanwhile kept the most likely sentence and thereby achieved the same accuracy as Viterbi. This reflects a fundamental underpinning of why beam search works: the least likely tag combinations can be discarded with little effect on the overall accuracy.

The tradeoff here is time. The Viterbi algorithm took 24.5 second to run on my machine. When compared to the smaller beam sizes of 1, 2, and 3 we can see that Beam Search ran faster. This is expected, because beam search discards the least likely tag combinations and we no longer iterate through them. However, when the beam size is 4 or more, the beam search algorithm took longer to run than Viterbi. This means that the overhead of sorting through the beam which occurs at every word iteration slowed the algorithm enough that it no longer showed any gains in time efficiency. This means that after a beam size of 4 or more we cannot see any benefit in using beam search over Viterbi. Interestingly, the time and efficacy tradeoff have quite near each other at beam size of 4 and 5 respectively.

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?

The circumstances under which beam search is more effective, effective is defined here as runs faster or is more accurate, is when there are many states and we don’t see a large drop in accuracy with smaller beam sizes.

In this problem, our tags were our states, and there were relatively few of them (8). However, we still saw that smaller beam sizes ran faster than the Viterbi Algorithm. This is hardly surprising when you consider that Viterbi runs in O(n t2) where t is the number of tags and n is the number of words. Comparatively, beam search runs in O(n t k log(k)) where t is the number of tags, n is the number of words, and k is the size of the beam. Thus, when k << t we would expect to see beam search preform much more efficiently than Viterbi. However when K is close to T we would expect to see no gains.

This also assumes that there is little accuracy loss between beam search and Viterbi. Based on the previous results this is not a bad assumption, but we would expect a small decrease in accuracy.