NLP HW3

2. Decoding Katakana to English Phonemes

1) Trigram Viterbi Algorithm

Based on Prof’s algorithm in the lecture slides, we implemented the trigram viterbi algorithm.

The input to the algorithm is a sequence of a japanese pronunciation, katakana, for a word, a japanese pronunciation for an english word (emission tags), and english pronunciation (transition tags).
Then we initialize the first best score as 1 and set the 1st steps as 0 for best_prediction(backtrace).

```
For the length of pronunciation of a word:
    For each trigram tag:
        Calculate the best(max) score
        Compare the with previous scores:
            Set it back (backtrace/backpointers): store backpointer values at each step.
```

Return the best sequence with the best prediction(the highest probability sequence)

2) Define Subproblem and recurrence relations : the best score calculation and backtrace

Best_score: calculate the highest probability for a sequence

Backpointers: store backpointersvalues at each step, which record the previous state which leads to the highest scoring sequence ending in transition tags(trigram) at the position of each tag.

3) Complexity Analysis

The running time for the algorithm is O(n * T^3), where n is the length of the pronunciation sequence, and T^3 is cubic in the number of tags.

4) Running Time:

```
python decode.py epron.wfsa epron-jpron.wfst  0.33s user 0.04s system 94% cpu 0.394 total
```

3. K-Best Output

```
python kbest.py epron.wfsa epron-jpron.wfst  1.50s user 0.13s system 96% cpu 1.695 total
```