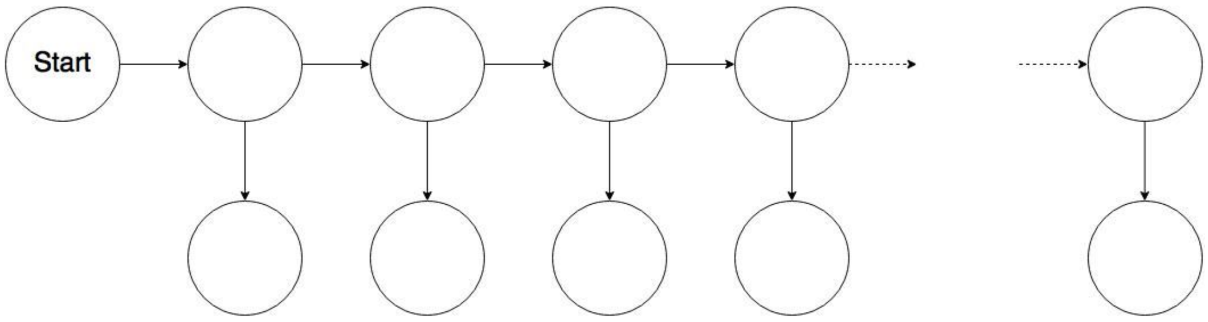
1. Graphical Model:



2. decode.cpp/decode.py:
    a. I first wrote decode.py, but it was really slow. Then I wrote it in C++.
    b. Compile and run decode.cpp:
        i. g++ -O2 decode.cpp -o decode
        ii. ./decode
    c. Run decode.py
        i. python decode.py
        ii. where python is Python2
3. Viterbi Algorithm (Dynamic Programming)
    a. Use DP to restore which node (word) has the maximal probability.
    b. The formula: (alpha = {alphabets, digits and space})

$$P(this\,layer) = \max\left(P(previous\ layer) * P(alpha|previous\ alpha) * P(observed\ alpha|alpha)\right)$$

    c. When implementing, I added log to all the probability since it may be very small for the probability.
4. Result:
    a. Both the results generated by the two codes are same.
    b. See the pred.txt file
5. Time:
    a. decode.py: ~2500s
    b. decode.cpp: ~13s