CSCI 544: Applied Natural Language Processing

HW2

Submission by: Sairaj Pokale (USC ID: 8392909073)

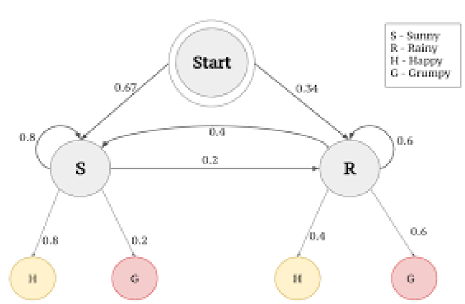
- **Objective:** Implement the Hidden Markov Model for Parts of Speech tagging on the Wall Street Journal section of the Penn Treebank dataset. The test.json file contains the raw sentences that you need to predict the part-of-speech tags.

- **Questions:**

  - What threshold value did you choose for identifying unknown words for replacement?
    - Words with frequency 2 or less are marked as unknown words.
  - What is the overall size of your vocabulary, and how many times does the special token ”<unk>” occur following the replacement process?
    - Size of vocab : 43193 Total occurrences of <unk> is : 32537
  - How many transition and emission parameters in your HMM?
    - Transition: 1351, Ideally it should hold T×T values but I have not stored the parameters having the values as 0.
    - Emission: 50320, ideally it should hold T×W values but I have not stored the parmeters having the values as 0.
  - What is the accuracy of Greedy Algorithm on the dev data?
    - Accuracy of Greedy on dev_data: 92.67728128225366%
  - What is the accuracy of Viterbi Algorithm on the dev data?
    - Accuracy of Viterbi on dev_data: 94.36585513933579%

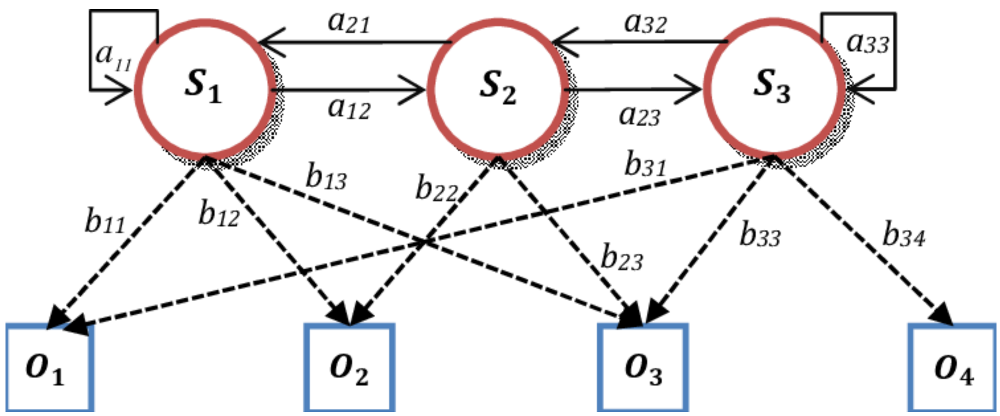
- **Markov Assumption**

  If we want to predict the future in the sequence, all that matters is the current state. The states before the current state have no impact on the future except via the current state.



- **Hidden Markov Model:**

  Hidden Markov models are probabilistic frameworks where the observed data are modeled as a series of outputs generated by one of several (hidden) internal states.



- **Greedy Algorithm for Decoding:**

Algorithm 1 Greedy Decoding for a sentence

Input:

hmm with

1. $\pi(s_i)$ as an element in initial state vector $\in \mathbb{R}^{|S|}$
2. $t(s_i|s_j)$ as an element in transition state matrix $\in \mathbb{R}^{|S|\times|S|}$
3. $e(w_i|s_i)$ as an element in emission matrix $\in \mathbb{R}^{|W|\times|S|}$

where S is a set of all tags and W is a set of all words.

sentence $\leftarrow \{w_1, w_2, ..., w_T\}$

Output: $\{y_1, y_2, ..., y_T\}$ (A list of tags)

function DECODE($\{w_1, w_2, ..., w_T\}$)

$y_1 \leftarrow \underset{s\in S}{argmax}\ \pi(s) * e(w_1|s)$

for $i \leftarrow 2$ to T **do**

$y_i \leftarrow \underset{s\in S}{argmax}\ t(s|y_{i-1}) * e(w_i|s)$

end for

return $\{y_1, y_2, ..., y_T\}$

end function

- Viterbi Algorithm for Decoding:

function *VITERBI* $(O, S, \Pi, Y, A, B) : X$
    for each state $i = 1, 2, \ldots, K$ **do**
        $T_1[i, 1] \leftarrow \pi_i \cdot B_{iy_1}$
        $T_2[i, 1] \leftarrow 0$
    end for
    for each observation $j = 2, 3, \ldots, T$ **do**
        for each state $i = 1, 2, \ldots, K$ **do**
            $T_1[i, j] \leftarrow \max_k \left(T_1[k, j-1] \cdot A_{ki} \cdot B_{iy_j}\right)$
            $T_2[i, j] \leftarrow \arg\max_k \left(T_1[k, j-1] \cdot A_{ki} \cdot B_{iy_j}\right)$
        end for
    end for
    $z_T \leftarrow \arg\max_k \left(T_1[k, T]\right)$
    $x_T \leftarrow s_{z_T}$
    for $j = T, T-1, \ldots, 2$ **do**
        $z_{j-1} \leftarrow T_2[z_j, j]$
        $x_{j-1} \leftarrow s_{z_{j-1}}$
    end for
    return X
end function

- **References:**

    - https://math.stackexchange.com/questions/3605188/hidden-markov-model-understanding-viterbi-algorithm
    - https://towardsdatascience.com/markov-and-hidden-markov-model-3eec42298d75
    - https://web.stanford.edu/~jurafsky/slp3/A.pdf