NLP | תרגיל 2

3 בדצמבר 2022

שאלה 1

1. (10 pts) Consider this (toy) biological setup:
A cell can be in one of two states - H, for high GC-content, and L for low GC. On each time step the cell produces one nucleotide, A,C,T or G, and might also change its state. The probability of changing from state H to L is 0.5, and from state L to H is 0.4.
In state H the probabilities for producing nucleotides are 0.2 for A, 0.3 for C, 0.3 for G and 0.2 for T. In L the probabilities are 0.3 for A, 0.2 for C, 0.2 for G and 0.3 for T.
Consider the nucleotide sequence $S = ACCGTGCA$. Use the Viterbi algorithm to find the best state-sequence and calculate the probability of S given this state-sequence. Assume the previous state before S was H.

תחילה נחשב את ההסתברויות של q ו e:

:$transition\ probabilty$

$$q(H|H) = 0.5$$
$$q(L|H) = 0.5$$
$$q(L|H) = 0.4$$
$$q(L|L) = 0.6$$

: $emition\ probability$

$e(A|H) = 0.2$ ו $e(A|L) = 0.3$

$e(C|H) = 0.3$ ו $e(C|L) = 0.2$

$e(G|H) = 0.3$ ו $e(G|L) = 0.2$

$e(T|H) = 0.2$ ו $e(T|L) = 0.3$

כעת נגדיר טבלת תכנון דינאמי ונמלא אותה באמצעות הפונקציה של אלגוריתם $vertini$: - נסמן ב box את הmax בכל שלב

מקרה בסיס:

$$T(0, *, H) = 1$$

נתחיל למלא:

$$\boldsymbol{k = 1}$$

$$T(1, H, H) = T(0, *, H) * q(H|H) * e(A|H) = 0.1$$

$$\boxed{T(1, H, L) = T(0, *, H) * q(L|H) * e(A|L) = 0.15}$$

$$\boldsymbol{k = 2}$$

$$T(2, H, L) = T(1, H, H) * q(L|H) * e(C|L) = 0.01$$

$$\boxed{T(2, L, L) = T(1, H, L) * q(L|L) * e(C|L) = 0.018}$$

$$T(2, H, H) = T(1, H, H) * q(H|H) * e(C|H) = 0.015$$

$$\boxed{T(2, L, H) = T(1, H, L) * q(H|L) * e(C|H) = 0.018}$$

$$\boldsymbol{k = 3}$$

$$T(3, H, L) = T(2, L, H) * \underbrace{q(L|H)}_{0.5} * \underbrace{e(C|L)}_{0.2} = \frac{18}{10000}$$

$$T(3, L, L) = T(2, L, L) * \underbrace{q(L|L)}_{0.6} * \underbrace{e(C|L)}_{0.2} = \frac{27}{12500}$$

$$\boxed{T(3,H,H) = T(2,L,H) * \underbrace{q(H|H)}_{0.5} * \underbrace{e(C|H)}_{0.3} = \frac{27}{10000}}$$

$$T(3,L,H) = T(2,L,L) * \underbrace{q(H|L)}_{0.5} * \underbrace{e(C|H)}_{0.3} = \frac{27}{12500}$$

$$\boldsymbol{k = 4}$$

$$\boxed{T(4,H,H) = T(3,H,H) * \underbrace{q(H|H)}_{0.5} * \underbrace{e(G|H)}_{0.2} = \frac{81}{200,000}}$$

$$T(4,H,L) = T(3,H,H) * \underbrace{q(L|H)}_{0.5} * \underbrace{e(G|L)}_{0.3} = \frac{27}{100,000}$$

$$T(4,L,L) = \underbrace{T(3,L,L)}_{=T(3,L,H)} * \underbrace{q(L|L)}_{0.6} * \underbrace{e(G|L)}_{0.2} = \frac{81}{312500}$$

$$T(4,L,H) = \underbrace{T(3,L,L)}_{=T(3,L,H)} * \underbrace{q(H|L)}_{0.4} * \underbrace{e(G|L)}_{0.3} = \frac{81}{312500}$$

$$\boldsymbol{k = 5}$$

$$\boxed{T(5,H,L) = T(4,H,H) * \underbrace{q(L|H)}_{0.5} * \underbrace{e(T|L)}_{0.3} = 6.075 * 10^{-5}}$$

$$T(5,L,L) = T(4,H,L) * \underbrace{q(L|L)}_{0.6} * \underbrace{e(T|L)}_{0.3} = 4.86 * 10^{-5}$$

$$T(5,H,H) = T(4,H,H) * \underbrace{q(H|H)}_{0.5} * \underbrace{e(T|H)}_{0.2} = 4.05 * 10^{-5}$$

$$T(5,L,H) = T(4,H,L) * \underbrace{q(H|L)}_{0.4} * \underbrace{e(T|H)}_{0.2} = 2.16 * 10^{-5}$$

$$\boldsymbol{k = 6}$$

$$T(6,H,H) = T(5,H,H) * \underbrace{q(H|H)}_{0.5} * \underbrace{e(G|H)}_{0.3} = 6.075 * 10^{-6}$$

$$T(6,H,L) = T(5,H,H) * \underbrace{q(L|H)}_{0.5} * \underbrace{e(G|L)}_{0.2} = 4.05 * 10^{-6}$$

$$\boxed{T(6,L,L) = T(5,H,L) * \underbrace{q(L|L)}_{0.6} * \underbrace{e(G|L)}_{0.2} = 7.29 * 10^{-6}}$$

$$\boxed{T(6,L,H) = T(5,H,L) * \underbrace{q(H|L)}_{0.4} * \underbrace{e(G|H)}_{0.3} = 7.29 * 10^{-6}}$$

$k = 7$

$$T(7,H,L) = T(6,L,L) * \underbrace{q(L|H)}_{0.5} * \underbrace{e(C|L)}_{0.2} = 7.29 * 10^{-7}$$

$$T(7,L,L) = T(6,L,L) * \underbrace{q(L|L)}_{0.6} * \underbrace{e(C|L)}_{0.2} = 8.748 * 10^{-7}$$

$$\boxed{T(7,H,H) = T(6,L,H) * \underbrace{q(H|H)}_{0.5} * \underbrace{e(C|H)}_{0.3} = 1.0935 * 10^{-6}}$$

$$T(7,L,H) = T(6,L,H) * \underbrace{q(H|L)}_{0.4} * \underbrace{e(C|H)}_{0.3} = 8.748 * 10^{-7}$$

$k = 8$

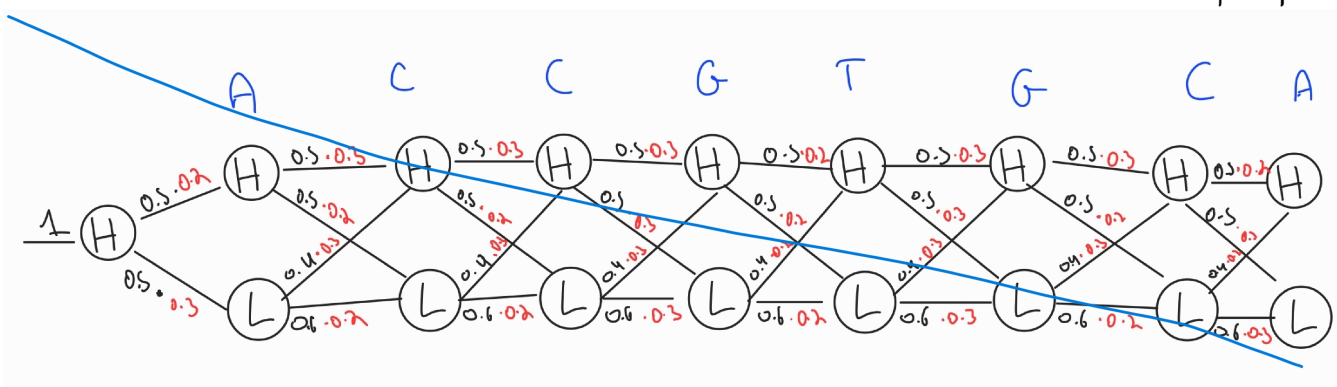
$$T(8,H,H) = T(7,H,H) * \underbrace{q(H|H)}_{0.5} * \underbrace{e(A|H)}_{0.2} = 1.0935 * 10^{-7}$$

$$\boxed{T(8,H,L) = T(7,H,H) * \underbrace{q(L|H)}_{0.5} * \underbrace{e(A|L)}_{0.3} = 1.64025 * 10^{-7}}$$

$$T(8,L,L) = T(7,L,L) * \underbrace{q(L|L)}_{0.6} * \underbrace{e(A|L)}_{0.3} = 1.57464 * 10^{-7}$$

$$T(8,L,H) = T(7,L,L) * \underbrace{q(H|L)}_{0.4} * \underbrace{e(A|H)}_{0.2} = 6.9984 * 10^{-8}$$

הרצף שקיבלנו הוא $HLLHHLHHL$



שאלה 2

2. (10 pts) In class we saw the trigram HMM model and the corresponding Viterbi algorithm. We will now make two main changes. First, we will consider a four-gram tagger, where p takes the form:

$$p(x_1 \cdots x_n, y_1 \cdots y_{n+1}) = \prod_{i=1}^{n+1} q(y_i|y_{i-3}, y_{i-2}, y_{i-1}) \prod_{i=1}^{n} e(x_i|y_i) \tag{1}$$

We assume in this definition that $y_0 = y_{-1} = y_{-2} = *$, where $*$ is the START symbol, $y_{n+1} = STOP$, and $y_i \in \mathcal{K}$ for $i = 1 \cdots n$, where $\mathcal{K}$ is the set of possible tags in the HMM.
Second, we consider a version of the Viterbi algorithm that takes as input **an integer** n (and not a sentence $x_1 \cdots x_n$ as we saw in class) and finds

$$\max_{y_1 \cdots y_{n+1}, x_1 \cdots x_n} p(x_1 \cdots x_n, y_1 \cdots y_{n+1})$$

for a four-gram tagger, as defined in Equation 1. $x_1 \cdots x_n$ may range over the values of some fixed vocabulary $\mathcal{V}$. Complete the following pseudo-code of this version of the Viterbi algorithm for this model . The pseudo-code must be efficient.

Input: An integer n, parameters $q(w|t,u,v)$ and $e(x|s)$.
Definitions: Define $\mathcal{K}$ to be the set of possible tags. Define $\mathcal{K}_{-2} = \mathcal{K}_{-1} = \mathcal{K}_0 = \{*\}$, and $\mathcal{K}_k = \mathcal{K}$ for $k = 1 \cdots n$. Define $\mathcal{V}$ to be the set of possible words.
Initialization: $\cdots$
Algorithm: $\cdots$
Return: $\cdots$

אתחול:

1. נגדיר בדומה להגדרה את קבוצת התיוגים האפשרים להיות:

   (א) $K_{-2}, K_{-1}, K_0 = \{*\}$

   (ב) לכל $k \in [1,n]$ נגדיר $K_k = K$

2. נאתחל טבלה בגודל $n * |K|^3$ ונאתחל $\pi(0,*,*,*) = 1$ (בדומה לאלגוריתם המקורי)

אלגוריתם:

1. נרוץ על $i \in [1, n]$

(א) נבחר את $T(k,u,v,w) = \max_{\substack{z,u,v,w \in K_{k-3},K_{k-2},K_{k-1}K_k \\ x \in V}} T(k-1,z,u,v) \cdot q(w|z,u,v) \cdot e(x|w)$

החזרה:

1. נחזיר את $\max_{u,v,w \in K_{k-2},K_{k-1}K_k} = \pi(n,u,v,w) \cdot q(STOP|u,v,w)$

חלק מעשי:

שאלה $B - ii$

ii. Using the test set, compute the error rate (i.e., $1-accuracy$) for known words and for unknown words, as well as the total error rate.

```
B-ii
known_error_rate = 0.11845412651864265
unknown_error_rate = 0.8018979833926453
error_rate = 0.17390049080935424
```

שאלה $C - iii$

iii. Run the algorithm from c)ii) on the test set. Compute the error rates and compare to the results from b)ii).

```
C-iii
known_error_rate = 0.17825722664432342
unknown_error_rate = 0.5336906584992342
error_rate = 0.22102450709415883
```

בהשוואה לסעיף $B - ii$ אנו מקבלים כי השימוש ב $Viterbi\ algorithm$ מעלה את השגיאה עבור $know\ words$ ו $error\ rate$ כללי

אך לעומת זאת הוא משפר את השגיאה עבור $unkown\ words$ בצורה משמעותית

שאלה $D - ii$

ii. Using the new probabilities, run the algorithm from c)ii) on the test set. Compute the error rates and compare to the results from b)ii) and c)iii).

```
known_error_rate = 0.1543778801843318
unknown_error_rate = 0.42725880551301687
error_rate = 0.18721208770960016
```

השימוש ב $Add\ one\ smoothing$ לעומת סעיף $b-ii$ העלה את השגיאה עבור $know\ words$ והעלה במעט את ה $error\ rate$ הכללי

אך עבור ה $unkown\ words$ קיבלנו שיפור משמעותי

השימוש ב $Add\ one\ smoothing$ לעומת $vertidi$ ללא ה $smooting$ הוריד את השגיאה עבור $know\ words$ וה $error\ rate$ הכללי

כנל עבור ה $unkown\ words$ קיבלנו שיפור קל

ה$error\ rate$ הכללי דומה ל $B-ii$ וטוב יותר מהטעות ב $C-iii$ ו $D-ii$

שאלה $E-ii$

ii. Using the pseudo-words as well as maximum likelihood estimation (as in c)i)), run the Viterbi algorithm on the test set. Compute the error rates and compare to the results from b)ii), c)iii) and d)ii).

```
E-ii
known_error_rate = 0.1510365251727542
unknown_error_rate = 0.46023166023166023
error_rate = 0.18949289281598156
```

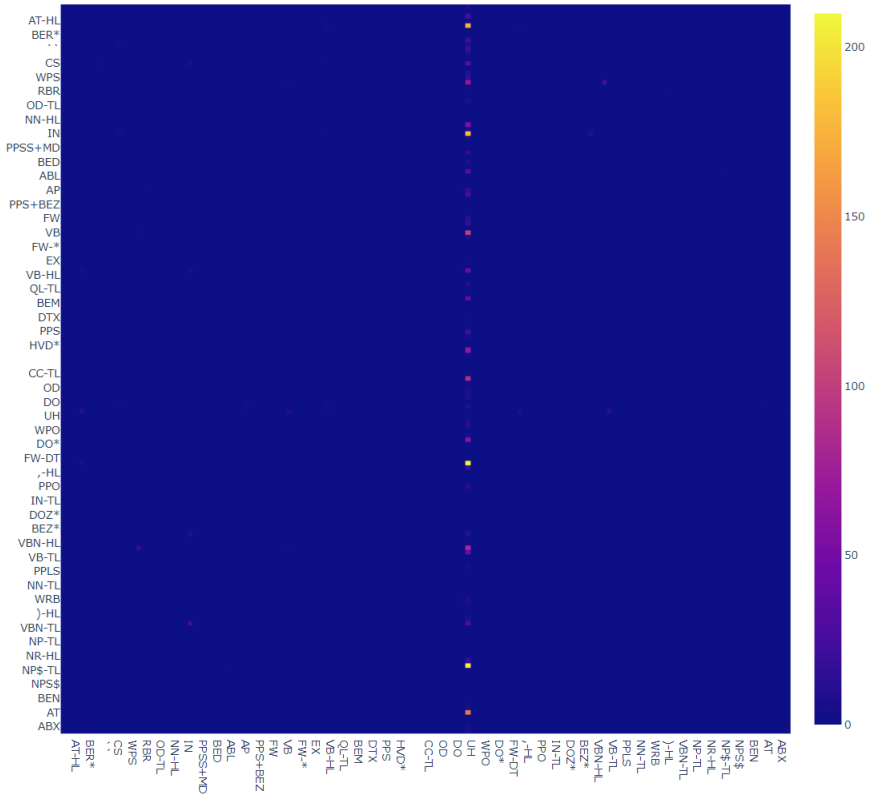
בסעיף זה קיבלנו שיפור משמעותי ב $known\ words\ error$ לעומת סעיף $C-iii$ ומס' טעויות דומה ל $D-ii$ אך הטעות יותר גדולה מסעיף $B-ii$

ב$unkown\ error\ rate$ קיבלנו שגיאה הכי נמוכה מכל הסעיפים למעט מסעיף $D-ii$

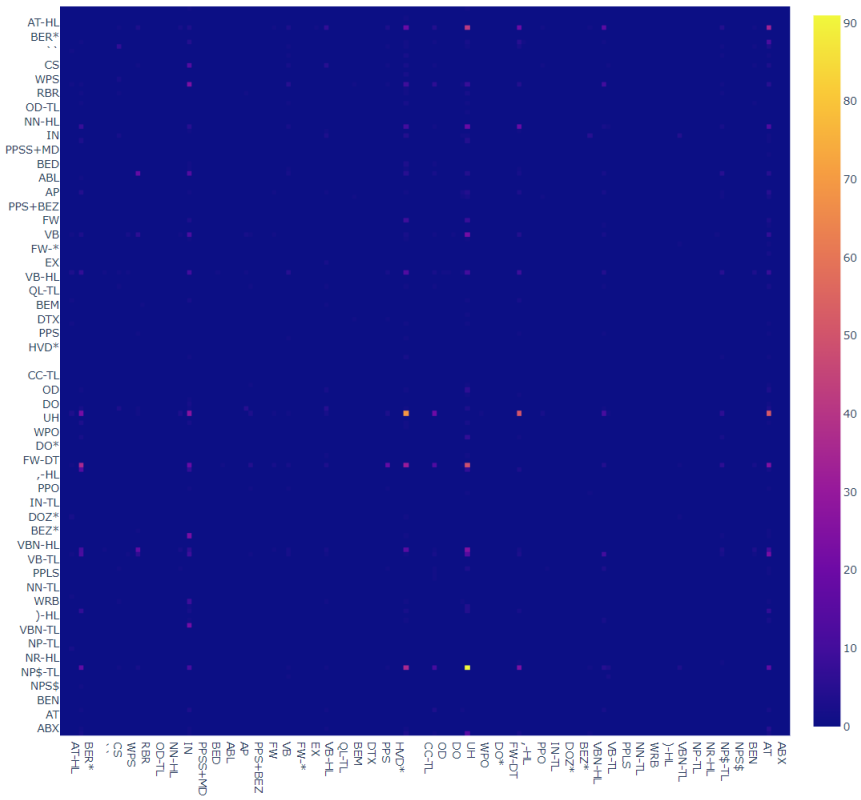
שאלה $E-iii$

iii. Using the pseudo-words as well as Add-One smoothing (as in d)i)), run the Viterbi algorithm on the test set. Compute the error rates and compare to the results from b)ii), c)iii), d)ii) and e)ii). For the results obtained using both pseudo-words and Add-One smoothing, build a confusion matrix and investigate the most frequent errors. A confusion matrix is an $|\mathcal{K}|$ over $|\mathcal{K}|$ matrix, where the (i, j) entry corresponds to the number of tokens which have a true tag i and a predicted tag j.

$C-ii$

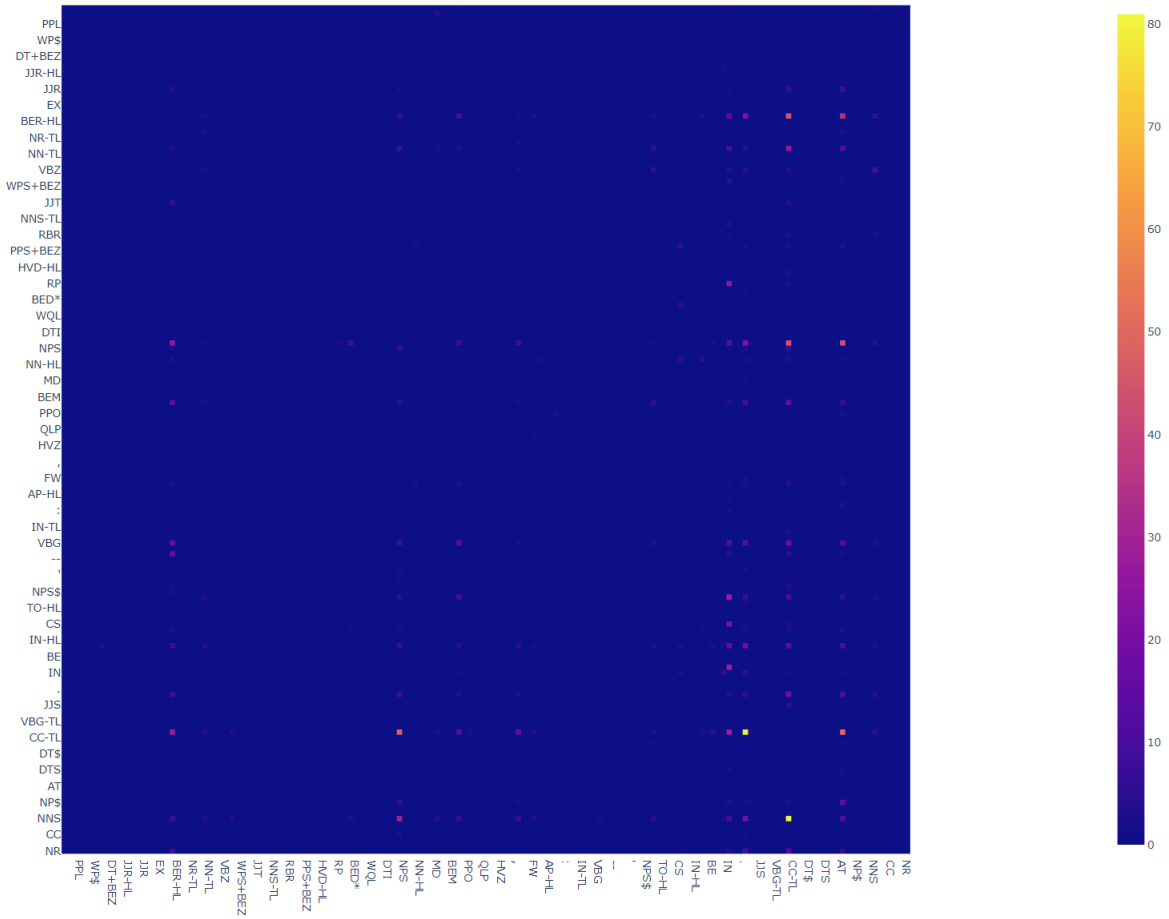


נבחין כי בריצת האלגוריתם $viterbi$ מסעיף $c-ii$ אנו מקבלים שרוב הטעויות נובעות מתיוגים בודדים $D-ii$



לאחר השימוש ב $add\ one\ smoothing$ שהטעויות בטווח גדול יותר של תיוגים בגלל הוספת הקבוע למילים שלא ראינו ב$train\ set$

$E - ii$



בשימוש בשתי המתודות ביחד - *pseudo words* ו *add one smoothing* קיבלנו עליה ב*unkown words* ולכן במטריצה רואים מעט יותר טעויות