NLP - ex2 - Practical Part

Guy Kornblit, 308224948; Mohamad Salama, 318983384

1. **Data:**

After loading the raw data, we considered only the prefix of each tag, before a '+' or '-' sign, resulting:

	# Sentences	# Words	# Wordforms	# Unique Tags
Train	4161	90,538	13,576	98
Test	462	10,016	2,826	79

In addition, we measured 2,008/2,826 wordforms and 78/79 tags from test that appeared in the train set.

2. **Models Comparison.**

In all the following HMM methods, we manually inserted smoothing with 1e-20 for all zero probabilities (transitions and emissions).

Model	**Error rate for Known Word**	**Error Rate for Unknown Words**	**Total Error Rate**
Baseline MLE	0.194	0.75	0.258
Base HMM	0.122	0.123	0.122
HMM-Add-1 Smoothing	0.130	0.134	0.130
HMM with Pseudo-words	0.127	0.123	0.126
HMM-Add-1 Smoothing with Pseudo-words	0.135	0.129	0.134

Generally, we can see that HMM variations preform rather similarly, and outperforms the MLE baseline model.

There is a major difference between baseline to the other models with respect to the unknown words error rate, that because we handle unknown words entirely different using the Viterbi algorithm - we just assume that the emission probability for that word is 1 for the default POS tag 'NN' and zero otherwise, same as baseline.

But, considering the back_pointers_table[k] where k is the location of the unknown word, we updated for every POS tag in the row the previous tag that maximizes the route and the transition. i.e

$$argmax_t\{transitions * \pi[k-1].\max(axis = 1)\}$$

Thus, we kept a probability model for these unknown words instead of tagging them in a deterministic way.

3. Pseudo words creation:

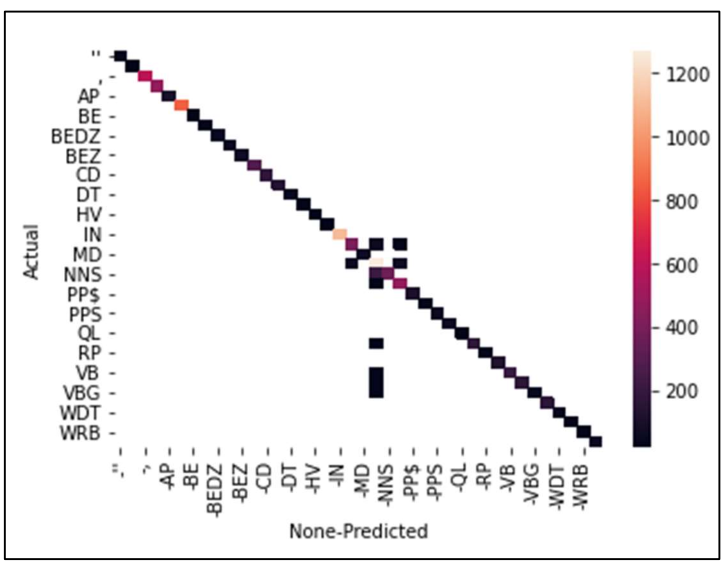
We chose threshold=2 to categorize low frequency training words by the following statistics:

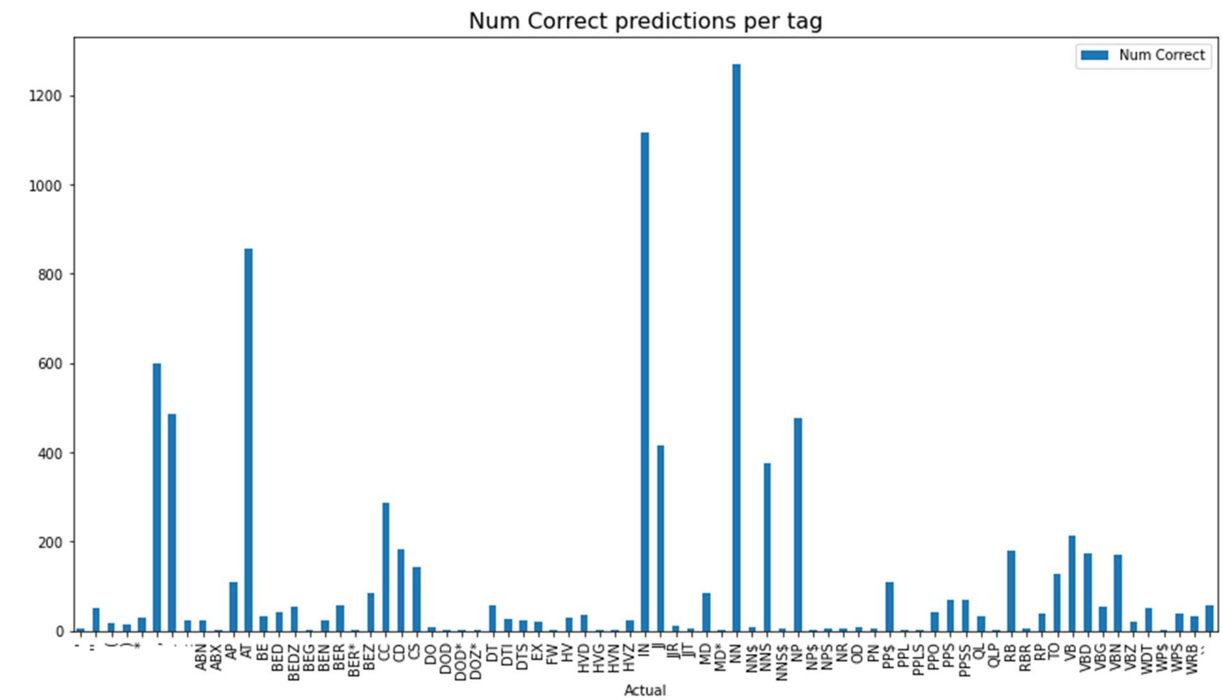
```
for t in [1,2,3,5, 20, 50]:
    print("Threshold ", t, "below: ", (word_freq.freq <= t).sum(), ". above: ",(word_freq.freq > t).sum())
```

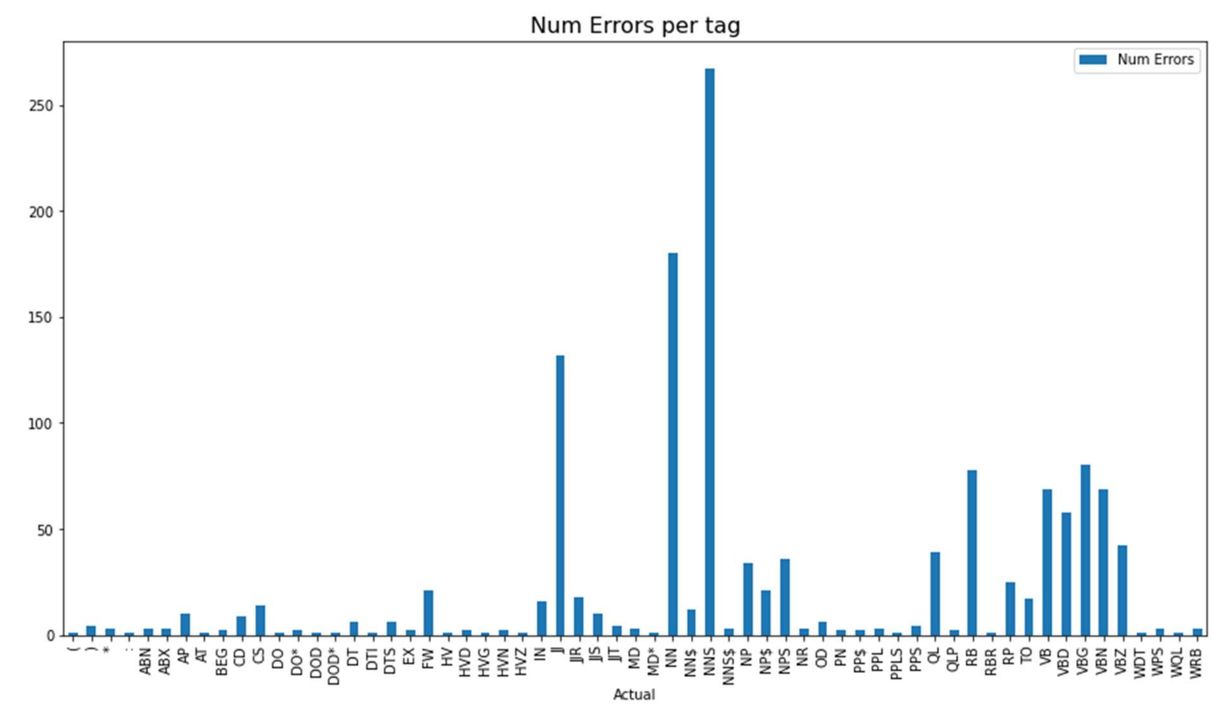
```
Threshold  1 below:  7398 . above:  6178
Threshold  2 below:  9510 . above:  4066
Threshold  3 below:  10537 . above:  3039
Threshold  5 below:  11610 . above:  1966
Threshold  20 below:  13101 . above:  475
Threshold  50 below:  13432 . above:  144
```

4. Error analysis:

Due to the sparsity of the confusion matrix, we decided to plot it only for the top 50 values of joint occurrences of tags. In addition, we show the number of error and correct predictions for every actual label.




Num Correct predictions per tag
Num Correct
0
200
400
600
800
1000
1200
Actual


Num Errors per tag
Num Errors
0
50
100
150
200
250
Actual

גיל הורנפליט
308224948

מוחמד סלאמה
318983384

* * * * *

תרגיל 2

שאלה 1

משתי ההנחות אפשר להסיק:

1) מהנחה מס' (1) נקבל שעבור התיוגים $y_1,y_2,\dots,y_n$

$$\mathbb{P}(y_1,\dots,y_n)=\prod_{i=1}^{n} P(y_i|y_{i-m},y_{i-m+1},\dots,y_{i-1})$$

וזו בדיוק הנוסחה שמייצגת את התגיות כשרשרת מרקוב מסדר m.

2) מהנחה מס' (2) נקבל שעבור הקלט $x_1,\dots,x_n$ ותיוגים $y_1,\dots,y_n$

$$P(x_1,\dots,x_n)=\prod_{i=1}^{n} P(x_i|y_i)=\prod_{i=1}^{n} e(x_i|y_i)$$

כעת, לפי ש y לא תלויה ב x, וגם ההסתברות של x_i בהינתן y_i לא תלויה ב $x_j|y_j$, אז ההסתברות המשותפת של משפט ושל סדרת התיוגים תהיה מכפלת ההסתברויות כלומר:

$$\mathbb{P}(x_1,x_2,\dots,x_n,y_1,y_2,\dots,y_n)=\prod_{i=1}^{n} P(y_i|y_{i-m},y_{i-m+1},\dots,y_{i-1})\cdot\prod_{i=1}^{n} e(x_i|y_i)$$

$$=\prod_{i=1}^{n} P(y_i|y_{i-m},y_{i-m+1},\dots,y_{i-1})\cdot e(x_i|y_i)$$

וזו הנוסחה של HMM עבור שרשרת מרקובית מסדר m.

שאלה 2

emission

	H	L
A	0.2	0.3
C	0.3	0.2
G	0.3	0.2
T	0.2	0.3

transsion

	H	L
H	0.5	0.5
L	0.4	0.6

input ⇒ S= ACCGTGCA

בכל שלב $k=1,\dots,8$ נחשב את הטבלה לפי הנוסחה: $V\in\{H,L\}$

$$\pi(K,V) = \max_{u\in S_{k-1}} \{\pi(k-1,u)\cdot q(v|u)\cdot e(x_k|v)\}$$

K	X_k	BP-H	H	BP-L	L
1	A	H	0.1	H	0.12
2	C	L	0.018	L	0.0144
3	C	H	$2.7\cdot e^{-3}$	L	$1.728\cdot e^{-3}$
4	G	H	$4.05\cdot e^{-4}$	H	$2.16\cdot e^{-4}$
5	T	H	$4.05\cdot e^{-5}$	H	$4.86\cdot e^{-5}$
6	G	L	$7.29\cdot e^{-6}$	L	$5.832\cdot e^{-06}$
7	C	H	$1.0935\cdot e^{-6}$	L	$6.9984\, e^{-7}$
8	A	H	$1.0935\cdot e^{-7}$	H	$1.3122\, e^{-7}$

אז נקבל שהתיוגים הנכונים הם:

S= L H H H L H H L

וההסתברות של S בהינתן הרצף הזה היא

$S= 1.2\cdot e^{-31}$

וחשבנו את זה לפי הנוסחה

$$P(x_1,x_2,\dots,x_8,y_1,\dots,y_8) = \prod_{i=1}^{8} p(y_i|y_{i-1}) \prod_{i=1}^{8} e(x_i|y_i)$$

כאשר $X=(A,C,C,G,T,G,C,A)$ $Y=(L,H,H,H,L,H,H,L)$

$y_0=H$

שאלה 3

נשנה את האלגוריתם :-

initialization: Set $\pi(0,*,*,*)=1$

Algorithm: For $k=1,\dots,n$

For $u\in K_{k-2}, v\in K_{k-1}, t\in K_k$:

$$\pi(k,u,v,t)=\max_{w\in K_{k-3}}\left(\pi(k-1,w,u,v)\cdot q(t|w,u,v)\cdot \max_{x\in V}(e(x|t))\right)$$

Return: $$\max_{u\in K_{n-2}, v\in K_{n-1}, t\in K_n}\left(\pi(n,u,v,t)\cdot q(STOP|u,v,t)\right)$$

אז לפי שהמילים עצמם אין ביניהם תלות, אז מספיק לקחת את המילה שנותנת הסתברות מקסימלית בהינתן תיוג $t\in K$, ולכן נקח $\max_{x\in V}(e(x|t))$, ושאר האלגוריתם כפי שלמדנו אבל עכשיו four-gram.