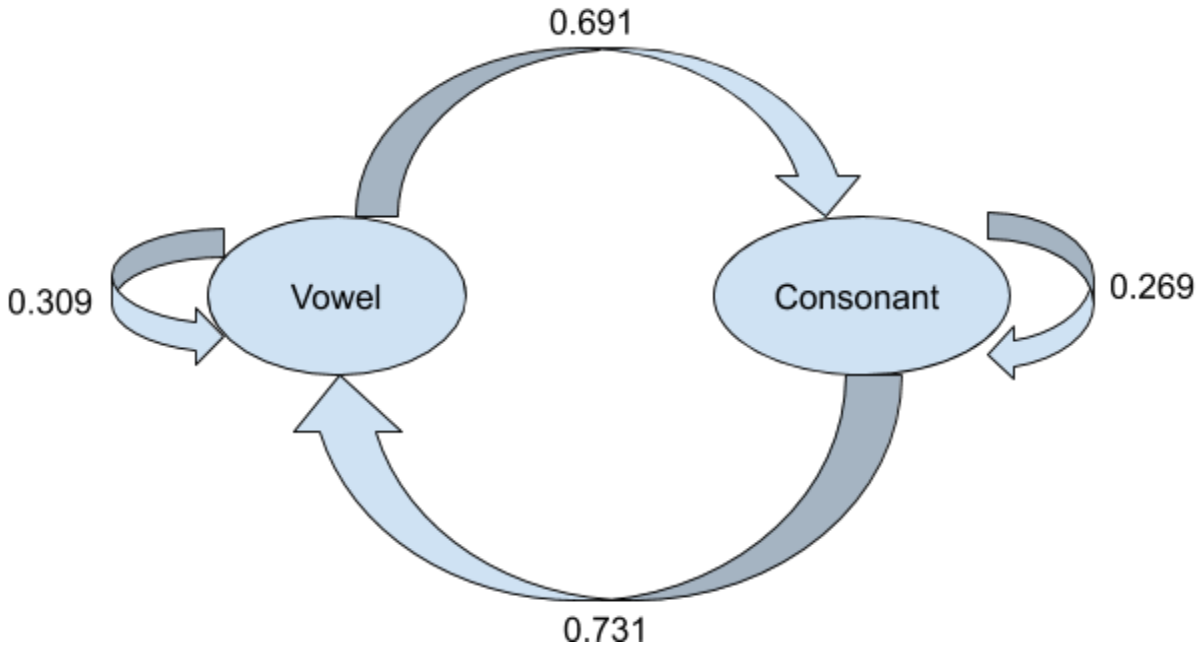
VIPIN KUMAR SETH
111701030

TASK -2

Natural 2-state Model



Transition prob :

[0.30898021, 0.69101979]

[0.7306307 , 0.2693693]

Here the transition probability is calculated on the training data, reading it in sequence and noting the transition.

The emission probability is

	State-0	State-1
A	0.122	-----
B	-----	0.024
C	-----	0.040
D	-----	0.076
E	0.204	-----
F	-----	0.044
G	-----	0.026
H	-----	0.113
I	0.111	-----
J	-----	0.002
K	-----	0.007
L	-----	0.065
M	-----	0.045
N	-----	0.117
O	0.127	-----
P	-----	0.023

Q ----- 0.002
R ----- 0.104
S ----- 0.106
T ----- 0.141
U 0.043 -----
V ----- 0.022
W ----- 0.040
X ----- 0.002
Y 0.034 -----
Z ----- 0.000
 # 0.359 -----
(Here # is space)

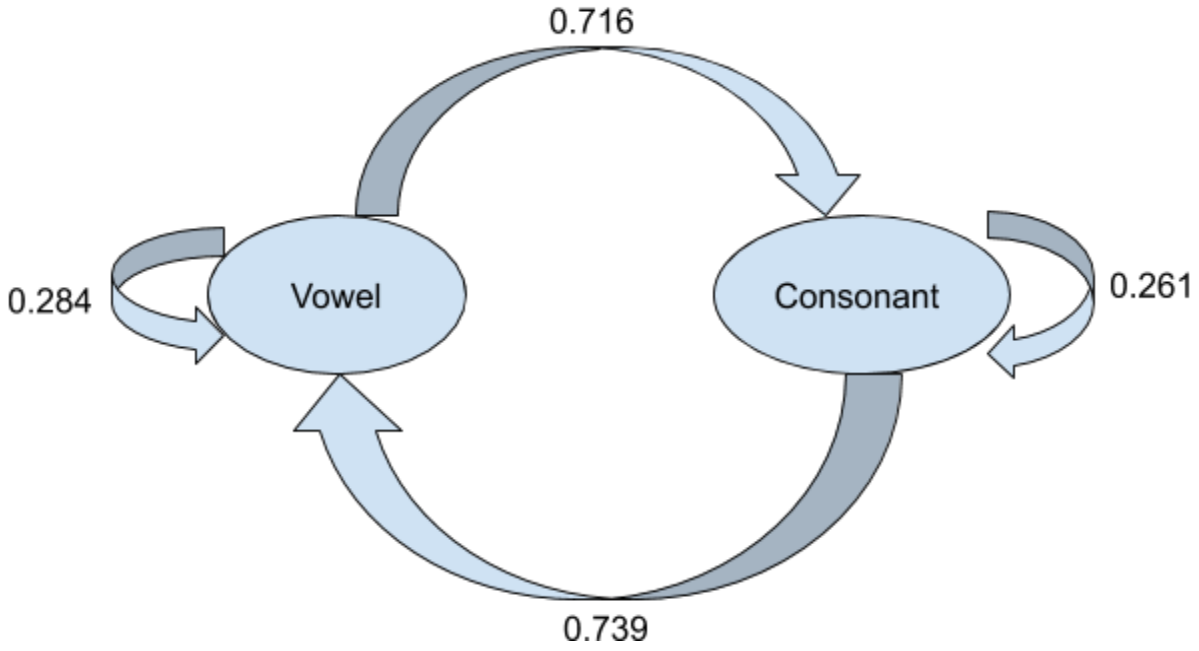
By the emission probability, the 7 most likely characters in Vowels are (in decreasing order):
Space
E
O
A
I
U
Y

The 7 most likely characters in Consonants are (in decreasing order):
T
N
H
S
R
D
L

TASK -3

After Training on training data
Transition Prob :

[0.28443182 0.71556818]
[0.7393169 0.2606831]



stationary distribution :

[0.50816172 0.49183828]

The emmision prob of this trained model :

	State-0	State-1
A	**0.123**	**0.000**
B	**-----**	**0.024**
C	**0.000**	**0.039**
D	**-----**	**0.075**
E	**0.206**	**-----**
F	**-----**	**0.044**
G	**0.002**	**0.024**
H	**0.009**	**0.103**
I	**0.112**	**-----**
J	**-----**	**0.002**
K	**-----**	**0.007**
L	**0.011**	**0.053**
M	**-----**	**0.045**
N	**-----**	**0.115**
O	**0.129**	**-----**
P	**0.002**	**0.021**
Q	**-----**	**0.002**
R	**-----**	**0.103**
S	**-----**	**0.104**
T	**0.000**	**0.139**
U	**0.043**	**0.001**
V	**-----**	**0.021**

W ----- 0.040
X ----- 0.002
Y ----- 0.035
Z ----- 0.000
 # 0.363 0.000

So seeing the emission probabilities we can say that State 1 is Consonant and State 0 is Vowel which our model has discovered automatically(after training).
(Note the run the training again and again can alter the sates)
7 most likely characters in vowels

Space
E
O
A
I
U
L

7 most likely characters in consonants

T
N
S
H
R
D
L

Task -4
So now for comparing the model (Natural vs Trained) if calculated Log_prob (Obs | Model). This basically suggests how likely this observation you will observe (Here Observation is the training data).
Log_Prob of Trained one is = -73110.39468200509
Log_Prob of Natural one is = -73369.13156447651
Hence the Trained model is better.

Now initializing the model with natural model's params and training it.
So now the score i.e Log_prob is

Log_Prob of Trained one after initializing with natural params is = -73110.3997567153

So the score got increased hence the model doesn't degrade but it is better. Further, if we compare the number of steps taken by the initialized and non-initialized one then initialized took fewer steps to converge
This is evident from the printing of model.monitor_.
So this tells us that instead of giving some random initial params, giving useful params will help the model to converge fast.

Task -5
Now I evaluated the test data and got its Log_prob as -37218.440141262625.
Now to check the overfitting I compared it with a natural model. The Log_prob of the test data for the natural model is -37304.99995811899. This prob is quite close to the trained on which suggests the trained model has overfitting because it's giving higher prob than the natural model on the training data but giving similar score as natural model on the test data.