In [2]:

```
# Import Libraries
import numpy as np
```

In [3]:

```
# Load Text For Training
def load_text(filename):
    with open(filename, encoding='utf-8') as f:
        return f.read().lower()

data = load_text('data.txt')
# data = load_text('english_speech_2.txt')
```

In [4]:

```
# Prepare Markov chain transition table
def createTransitionTable(data, weight=4):
    T = {}
    for i in range(0, len(data)-weight):
        substr = data[i : i+weight]
        next_char = data[i+weight]
        if T.get(substr) is None:
            T[substr] = {}
            T[substr][next_char] = 1
        elif T[substr].get(next_char) is None:
            T[substr][next_char] = 1
        else:
            T[substr][next_char] += 1

    return T


T = createTransitionTable(data)
```

In [5]:

```
def changeTransitionTableToProb(T):
    for kx in T.keys():
        factor = (sum(T[kx].values()))
        for val in T[kx].keys():
            T[kx][val] = T[kx][val]/factor
    return T

T = changeTransitionTableToProb(T)
```

In [6]:

```
# Write prediction function
def predict(ctx, T):
    if T.get(ctx) is None:
        return " "
    sample_array = list(T[ctx].keys())
    sample_probab = list(T[ctx].values())

    return np.random.choice(sample_array, p=sample_probab)
```

In [104]:

```
# Sampling
```

In [8]:

```
# Generate
sentence = "कौन "
#sentence = "time"
# sentence = 'my d'
for i in range(35):
    char_returned = predict(sentence[-4:], T)
    sentence += char_returned

print(sentence)
```

कौन बोला, कौन बोला, कौन बोला
अपना time

In [9]:

```
# Text To Speech
from gtts import gTTS
import os
language = "en"
myobj = gTTS(text=sentence, lang=language, slow=False)
myobj.save("ML_RAP.mp3")

os.system("mpg321 ML_RAP.mp3")
```

Out[9]:

```
32512
```

In [ ]:

```

```

In [ ]:

```

```