IT 302

Assignment 4

Name: Niraj Nandish

Roll no.: 191IT234

```
!pip install snscrape
import snscrape.modules.twitter as sntwitter
import pandas as pd
import random
```

```
    Requirement already satisfied: snscrape in /usr/local/lib/python3.7/dist-packages (0.3.4)
    Requirement already satisfied: lxml in /usr/local/lib/python3.7/dist-packages (from snscrape) (4.2.6)
    Requirement already satisfied: requests[socks] in /usr/local/lib/python3.7/dist-packages (from snscrape) (2.23.0)
    Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.7/dist-packages (from snscrape) (4.6.3)
    Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests[socks]->sn
    Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests[socks]->snscrape
    Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests[socks]->sns
    Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (frc
    Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in /usr/local/lib/python3.7/dist-packages (from requests[socks]
```

a) Character, word and sentence generation for the Indian language Kannada

Text is stored in a text file with name `text.txt`

```
# Character generation
def markov_text_char():
    text_data = open('text.txt', 'r').read()
    text_data = ''.join([i for i in text_data if not i.isdigit()])
```

```
    index = 1
    markov_gen = {}
    word_count = int(input('Enter number of characters to generate: '))
    for character in text_data[index:]:
        key = text_data[index-1]
        if key in markov_gen:
            markov_gen[key].append(character)
        else:
            markov_gen[key] = [character]
        index += 1
    character1 = random.choice(list(markov_gen.keys()))
    message = character1.capitalize()
    while len(message.split(' ')) < word_count:
        character2 = random.choice(markov_gen[character1])
        character1 = character2
        message += ' ' + character2
    return message

# Word generation
def markov_text_word():
    text_data = open('text.txt', 'r').read()
    text_data = ''.join([i for i in text_data if not i.isdigit()]).replace('\n', ' ').split(' ')
    index = 1
    markov_gen = {}
    word_count = int(input('Enter number of words to generate: '))
    for character in text_data[index:]:
        key = text_data[index-1]
        if key in markov_gen:
            markov_gen[key].append(character)
        else:
            markov_gen[key] = [character]
        index += 1
    character1 = random.choice(list(markov_gen.keys()))
    message = character1.capitalize()
    while len(message.split(' ')) < word_count:
        character2 = random.choice(markov_gen[character1])
        character1 = character2
```

```
        message += ' ' + character2
    return message

# Sentence generation
def markov_text_sentence():
    text_data = open('text.txt', 'r').read()
    text_data = ''.join([i for i in text_data if not i.isdigit()]).replace('?', '.').split('.')
    index = 1
    markov_gen = {}
    word_count = int(input('Enter number of sentences to generate: '))
    for character in text_data[index:]:
        key = text_data[index-1]
        if key in markov_gen:
            markov_gen[key].append(character)
        else:
            markov_gen[key] = [character]
        index += 1
    character1 = random.choice(list(markov_gen.keys()))
    message = character1.capitalize()
    while len(message.split('\n')) < word_count:
        character2 = random.choice(markov_gen[character1])
        character1 = character2
        message += '\n' + character2
    return message


message_char = markov_text_char()
message_word = markov_text_word()
message_sentence = markov_text_sentence()
print('\nGenerated characters: ', message_char)
print('\nGenerated words: ', message_word)
print('\nGenerated sentences: ', message_sentence)
```

```
Enter number of characters to generate: 3
Enter number of words to generate: 3
Enter number of sentences to generate: 3

Generated characters:  ◌ೞ ರ ◌ೄ
```

```
Generated words:  ಈ ಸ್ಥಳದಲ್ಲಿ ಅಸ್ತಿತ್ವದ

Generated sentences:   ನಾನು ತುಂಬಾ ಸಂತೋಷವಾಗಿದ್ದೇನೆ, ನನ್ನ ಪ್ರಿಯ ಸ್ನೇಹಿತ, ಕೇವಲ ಶಾಂತ ಅಸ್ತಿತ್ವದ ಸೊಗಸಾದ ಅರ್ಥದಲ್ಲಿ ಲೀನವಾಗಿದ್ದೇನೆ, ನನ್ನ ಪ್ರತಿಭೆಯನ್ನು
 ನಾನು ಪ್ರಸ್ತುತ ಕ್ಷಣದಲ್ಲಿ ಒಂದೇ ಒಂದು ಹೊಡೆತವನ್ನು ಸೆಳೆಯಲು ಅಸಮರ್ಥನಾಗಿರಬೇಕು; ಮತ್ತು ಇನ್ನೂ ನಾನು ಈಗಿನಷ್ಟು ದೊಡ್ಡ ಕಲಾವಿದನಾಗಿರಲಿಲ್ಲ ಎಂದು ನಾನು ಭಾವಿಸುತ್ತೇ
 ಸುಂದರವಾದ ಕಣಿವೆಯು ನನ್ನ ಸುತ್ತಲೂ ಆವಿಯಿಂದ ತುಂಬಿರುವಾಗ ಮತ್ತು ಮೆರಿಡಿಯನ್ ಸೂರ್ಯನು ನನ್ನ ಮರಗಳ ತೂರಲಾಗದ ಎಲೆಗಳ ಮೇಲಿನ ಮೇಲ್ಮೈಯನ್ನು ಹೊಡೆದಾಗ
```

b) Crawl your WhatsApp text (Remove the media and personnel messages) from your WhatsApp group (OR Crawl 10,000 Tweets from the Twitter in your interested topic).

```
print("Scraping twitter...")
sentence = ""
for i,tweet in enumerate(sntwitter.TwitterSearchScraper('football since:2021-01-01 until:2021-11-01').get_items()):
  if i % 1000 == 0:
    print('Scraped {} tweets, {} tweets remaining to be scraped'.format(i, 10000-i))
  if i > 10000:
    break
  sentence+=tweet.content
print("Finished scraping twitter")

text_data = ''.join([i for i in sentence if not i.isdigit()]).replace("\n", " ").split(' ')
markov_gen = {}
index = 1

# Dictionary for one word
for character in text_data[index:]:
  key = text_data[index-1]
  if key in markov_gen:
    markov_gen[key].append(character)
  else:
    markov_gen[key] = [character]
  index += 1
# Dictionary for two words
for i in range(len(text_data)-2):
  key = tuple([text_data[i],text_data[i+1]])
  if key in markov_gen:
```

```
    markov_gen[key].append(text_data[i+2])
  else:
    markov_gen[key] = [text_data[i+2]]

# Dictionary for three words
for i in range(len(text_data)-3):
  key = tuple([text_data[i],text_data[i+1],text_data[i+2]])
  if key in markov_gen:
    markov_gen[key].append(text_data[i+3])
  else:
    markov_gen[key] = [text_data[i+3]]
```

```
Scraping twitter...
Scraped 0 tweets, 10000 tweets remaining to be scraped
Scraped 1000 tweets, 9000 tweets remaining to be scraped
Scraped 2000 tweets, 8000 tweets remaining to be scraped
Scraped 3000 tweets, 7000 tweets remaining to be scraped
Scraped 4000 tweets, 6000 tweets remaining to be scraped
Scraped 5000 tweets, 5000 tweets remaining to be scraped
Scraped 6000 tweets, 4000 tweets remaining to be scraped
Scraped 7000 tweets, 3000 tweets remaining to be scraped
Scraped 8000 tweets, 2000 tweets remaining to be scraped
Scraped 9000 tweets, 1000 tweets remaining to be scraped
Scraped 10000 tweets, 0 tweets remaining to be scraped
Finished scraping twitter
```

```
word = input("Enter 2 words: ")
sugg = word
word = word.split(' ')
sugg1 = random.choice(markov_gen[tuple([word[0],word[1]])])
word.append(sugg1)
sugg2 = random.choice(markov_gen[tuple([word[0],word[1],word[2]])])
print(sugg,"-->",sugg1,sugg2)
```

```
Enter 2 words: Football is
Football is --> a game.
```