EXPERIMENT NO 5

N - GRAM MODEL

IMPORTING LIBRARIES

In [2]:
```
from nltk.util import pad_sequence
from nltk.util import bigrams
from nltk.util import ngrams
from nltk.util import everygrams
from nltk.lm.preprocessing import pad_both_ends
from nltk.lm.preprocessing import flatten
from nltk.lm.preprocessing import padded_everygram_pipeline
```

READING THE DATA FROM TEXT FILE

In [4]:
```
file = open(
    '/DOCUMENTS/COLLEGE/CLASSES/EXPERIMENT_NO_5/language-never-random.txt',
    encoding='utf8')
text = file.read()
```

TOKENIZATION

In [6]:
```
from nltk import word_tokenize, sent_tokenize

word_tokenize(sent_tokenize("This is a foobar sentence. Yes it is.")[0])
```

Out[6]:
```
['This', 'is', 'a', 'foobar', 'sentence', '.']
```

In [7]:
```
# Tokenize the text.
tokenized_text = [
    list(map(str.lower, word_tokenize(sent))) for sent in sent_tokenize(text)
]
```

In [8]:
```
tokenized_text[0]
```

Out[8]:
```
['language',
 'is',
 'never',
 ',',
 'ever',
 ',',
 'ever',
 ',',
 'random',
 'adam',
 'kilgarriff',
 'abstract',
 'language',
 'users',
 'never',
 'choose',
 'words',
 'randomly',
 ',',
 'and',
 'language',
 'is',
 'essentially',
 'non-random',
 '.']
```

In [9]:
```
print(text[:500])
```

```
Language is never, ever, ever, random

ADAM KILGARRIFF

Abstract
Language users never choose words randomly, and language is essentially
non-random. Statistical hypothesis testing uses a null hypothesis, which
posits randomness. Hence, when we look at linguistic phenomena in cor-
pora, the null hypothesis will never be true. Moreover, where there is enough
data, we shall (almost) always be able to establish
```

READING THE DATA FROM A DATASET

In [10]:
```
import pandas as pd

df = pd.read_csv(
    '/DOCUMENTS/COLLEGE/CLASSES/EXPERIMENT_NO_5/Donald-Tweets!.csv')
df.head()
```

Out[10]:

	Date	Time	Tweet_Text	Type	Media_Type	Hashtags	Tweet_Id	Tweet_Url	twt_favourites_IS_THIS
0	16-11-11	15:26:37	Today we express our deepest gratitude to all ...	text	photo	ThankAVet	7.970000e+17	https://twitter.com/realDonaldTrump/status/797...	
1	16-11-11	13:33:35	Busy day planned in New York. Will soon be mak...	text	NaN	NaN	7.970000e+17	https://twitter.com/realDonaldTrump/status/797...	
2	16-11-11	11:14:20	Love the fact that the small groups of protest...	text	NaN	NaN	7.970000e+17	https://twitter.com/realDonaldTrump/status/797...	
3	16-11-11	2:19:44	Just had a very open and successful presidenti...	text	NaN	NaN	7.970000e+17	https://twitter.com/realDonaldTrump/status/796...	
4	16-11-11	2:10:46	A fantastic day in D.C. Met with President Oba...	text	NaN	NaN	7.970000e+17	https://twitter.com/realDonaldTrump/status/796...	

In [11]:
```
trump_corpus = list(df['Tweet_Text'].apply(word_tokenize))
```

PREPROCESSIMNG TASK

In [12]:
```
# Preprocess the tokenized text for 3-grams language modelling
n = 3
train_data, padded_sents = padded_everygram_pipeline(n, trump_corpus)
```

MODEL TRAINING

In [13]:
```
from nltk.lm import MLE

trump_model = MLE(n)
trump_model.fit(train_data, padded_sents)
```

In [14]:
```
from nltk.tokenize.treebank import TreebankWordDetokenizer

detokenize = TreebankWordDetokenizer().detokenize

def generate_sent(model, num_words, random_seed=42):
    """
    :param model: An ngram language model from `nltk.lm.model`.
    :param num_words: Max no. of words to generate.
    :param random_seed: Seed value for random.
    """
    content = []
    for token in model.generate(num_words, random_seed=random_seed):
        if token == '<s>':
            continue
        if token == '</s>':
            break
        content.append(token)
    return detokenize(content)
```

In [15]:
```
generate_sent(trump_model, num_words=20, random_seed=42)
```

Out[15]:
```
'call list for my press conference on taxes at 11AM at @ TwitterNYC http: //t.co/h4UOeJFW92'
```

In [16]:
```
generate_sent(trump_model, num_words=10, random_seed=0)
```

Out[16]:
```
'picked up the mess the U.S. is looking very bad'
```

In [17]:
```
generate_sent(trump_model, num_words=50, random_seed=10)
```

Out[17]:
```
'\\" @ about_life: @ realDonaldTrump Trumps new VA plan will help the veterans like no one is even higher than anticipated in radical Islamic terrorism, as her running mate @ realDonaldTrump Trumps new VA plan will help th
e veterans like no one is even higher than anticipated in'
```

In [18]:
```
print(generate_sent(trump_model, num_words=100, random_seed=52))
```

```
will win this thing! GET OUT TO VOTE Then``WE THE PEOPLE love you trump"Nice
```