Vidya Vikas Education Trust's
Universal College of Engineering, Kaman Road, Vasai-401208
Accredited B+ Grade by NAAC

Experiment 5

Name of the Student: - Sanket Belekar
Roll No.: 67 Batch: A3 Branch: TE COMPS-A
Date of Practical Performed: - Staff Signature with Date & Marks

Aim: Write a program to generate n-gram (bigram, trigram, etc.) of English Text.

Theory:

N-grams are continuous sequences of words or symbols or tokens in a document.
In technical terms, they can be defined as the neighboring sequences of items in a document.

They come into play when we deal with text data in NLP (Natural Language Processing) tasks.
● Thinking in the same lines, n-grams are classified into the following types, depending on the value that 'n' takes.

N= 1 then Unigram
N= 2 then Bigram
N= 3 then Trigram

Code:

N-grams Model:

```
import nltk
from nltk import ngrams
from collections import Counter
nltk.download('punkt')

def generate_ngrams(text, n):
    tokens = nltk.word_tokenize(text)
    ngrams_list = list(ngrams(tokens, n))
    return ngrams_list

def main():
  # Define the 'text' variable inside the main function
  text = " Natural language processor is used to perform the bigrams and
trigrams"

  #generate bigrams n=2
  Bigrams = generate_ngrams(text, 2)
  print("Bigrams:")
  print(Bigrams)
  print(Counter (Bigrams))

  #generate trigrams n=3
  Trigrams = generate_ngrams(text, 3)
  print("\nTrigrams:")
  print(Trigrams)
```

```
  print(Counter (Trigrams))

if___name___== "_main_":
    main()
```

Output:

```
Bigrams:
[('Natural', 'language'), ('language', 'processor'), ('processor', 'is'), ('is', 'used'), ('used', 'to'), ('to', 'perform'), ('perform', 'the'), ('the', 'bigrams'), ('bigrams', 'and'),
Counter({('Natural', 'language'): 1, ('language', 'processor'): 1, ('processor', 'is'): 1, ('is', 'used'): 1, ('used', 'to'): 1, ('to', 'perform'): 1, ('perform', 'the'): 1, ('the', 'b

Trigrams:
[('Natural', 'language', 'processor'), ('language', 'processor', 'is'), ('processor', 'is', 'used'), ('is', 'used', 'to'), ('used', 'to', 'perform'), ('to', 'perform', 'the'), ('perfor
Counter({('Natural', 'language', 'processor'): 1, ('language', 'processor', 'is'): 1, ('processor', 'is', 'used'): 1, ('is', 'used', 'to'): 1, ('used', 'to', 'perform'): 1, ('to', 'per
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Conclusion:

Thus, we have successfully studied and performed a program to generate n-gram (bigram,trigram, etc.) of English Text.