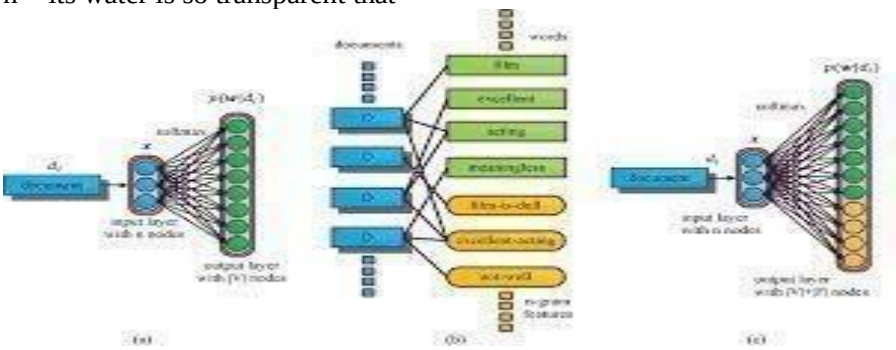
Name : Sarvagya Singh
SAPID : 60009200030
BATCH : K1

EXPERIMENT
NO 2

AIM: To make understand and learn the technique and use of N-Gram Model

THEORY:
Statistical language models, in its essence, are the type of models that assign probabilities to the sequences of words. In this article, we'll understand the simplest model that assigns probabilities to sentences and sequences of words, the n-gram
You can think of an N-gram as the sequence of N words, by that notion, a 2-gram (or bigram) is a two-word sequence of words like "please turn", "turn your", or "your homework", and a 3-gram (or trigram) is a three-word sequence of words like "please turn your", or "turn your homework"
Let's start with equation P(w|h), the probability of word w, given some history, h. For example, Here,
w = The wordgiven
h = its water is so transparent that



And, one way to estimate the above probability function is through the relative frequency count approach, where you would take a substantially large corpus, count the number of times you see its water is so transparent that, and then count the number of times it is followed by the. In other words, you are answering the question:Out of the times you saw the history h, how many times did the word w follow it Now, you can imagine it is not feasible to perform this over an entire corpus; especially it is of a significant a size.
This shortcoming and ways to decompose the probability function using the chain rule serves as the base intuition of the N-gram model. Here, you, instead of computing probability using the entire corpus, would approximate it by just a few historical words

The Bigram Model
As the name suggests, the bigram model approximates the probability of a word given all the previous words by using only the conditional probability of one preceding word. In other words, you approximate it with the probability: P(the | that)

And so, when you use a bigram model to predict the conditional probability of the next word, you are thus making the following approximation:
This assumption that the probability of a word depends only on the previous word is also known as **Markov assumption.**
Markov models are the class of probabilistic models that assume that we can predict the probability of some future unit without looking too far in the past.

Name : Sarvagya Singh
SAPID : 60009200030
BATCH : K1

You can further generalize the bigram model to the trigram model which looks two words into the past and can thus be further generalized to the N-gram model.

Data Set to Be Used

Lab Experiments to be performed in this Session: Step-by-step implementation of ngram language model

1. Basic pre-processing
2. Code to generate N-grams
3. Creating unigrams
4. Creating bigrams
5. Creating trigrams
6. Finding Frequency Distribution
7. Finding Probabilities for Bigram
8. Finding Next word for the given word using MLE (Maximum Likelihood Estimate)

Sarvayga Singh

> 60009200030 --- K1
>
> NLTK lab 2

In [1]:
```python
import numpy as np
import pandas as pd
import nltk
import re
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.util import ngrams
```

In [2]:
```python
# %%capture
nltk.download('all')
```

```
[nltk_data] Downloading collection 'all'
[nltk_data]    |
[nltk_data]    | Downloading package abc to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/abc.zip.
[nltk_data]    | Downloading package alpino to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/alpino.zip.
[nltk_data]    | Downloading package averaged_perceptron_tagger to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping taggers/averaged_perceptron_tagger.zip.
[nltk_data]    | Downloading package averaged_perceptron_tagger_ru to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping
[nltk_data]    |       taggers/averaged_perceptron_tagger_ru.zip.
[nltk_data]    | Downloading package basque_grammars to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping grammars/basque_grammars.zip.
[nltk_data]    | Downloading package bcp47 to /root/nltk_data...
[nltk_data]    | Downloading package biocreative_ppi to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/biocreative_ppi.zip.
[nltk_data]    | Downloading package bllip_wsj_no_aux to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping models/bllip_wsj_no_aux.zip.
[nltk_data]    | Downloading package book_grammars to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping grammars/book_grammars.zip.
[nltk_data]    | Downloading package brown to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/brown.zip.
[nltk_data]    | Downloading package brown_tei to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/brown_tei.zip.
[nltk_data]    | Downloading package cess_cat to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/cess_cat.zip.
[nltk_data]    | Downloading package cess_esp to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/cess_esp.zip.
[nltk_data]    | Downloading package chat80 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/chat80.zip.
[nltk_data]    | Downloading package city_database to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/city_database.zip.
[nltk_data]    | Downloading package cmudict to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/cmudict.zip.
[nltk_data]    | Downloading package comparative_sentences to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/comparative_sentences.zip.
[nltk_data]    | Downloading package comtrans to /root/nltk_data...
[nltk_data]    | Downloading package conll2000 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/conll2000.zip.
[nltk_data]    | Downloading package conll2002 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/conll2002.zip.
[nltk_data]    | Downloading package conll2007 to /root/nltk_data...
[nltk_data]    | Downloading package crubadan to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/crubadan.zip.
[nltk_data]    | Downloading package dependency_treebank to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/dependency_treebank.zip.
[nltk_data]    | Downloading package dolch to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/dolch.zip.
[nltk_data]    | Downloading package europarl_raw to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/europarl_raw.zip.
[nltk_data]    | Downloading package extended_omw to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    | Downloading package floresta to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/floresta.zip.
[nltk_data]    | Downloading package framenet_v15 to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/framenet_v15.zip.
[nltk_data]    | Downloading package framenet_v17 to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/framenet_v17.zip.
[nltk_data]    | Downloading package gazetteers to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/gazetteers.zip.
[nltk_data]    | Downloading package genesis to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/genesis.zip.
[nltk_data]    | Downloading package gutenberg to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/gutenberg.zip.
[nltk_data]    | Downloading package ieer to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/ieer.zip.
[nltk_data]    | Downloading package inaugural to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/inaugural.zip.
[nltk_data]    | Downloading package indian to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/indian.zip.
[nltk_data]    | Downloading package jeita to /root/nltk_data...
[nltk_data]    | Downloading package kimmo to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/kimmo.zip.
[nltk_data]    | Downloading package knbc to /root/nltk_data...
[nltk_data]    | Downloading package large_grammars to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping grammars/large_grammars.zip.
[nltk_data]    | Downloading package lin_thesaurus to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/lin_thesaurus.zip.
[nltk_data]    | Downloading package mac_morpho to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/mac_morpho.zip.
[nltk_data]    | Downloading package machado to /root/nltk_data...
[nltk_data]    | Downloading package masc_tagged to /root/nltk_data...
[nltk_data]    | Downloading package maxent_ne_chunker to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping chunkers/maxent_ne_chunker.zip.
[nltk_data]    | Downloading package maxent_treebank_pos_tagger to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping taggers/maxent_treebank_pos_tagger.zip.
[nltk_data]    | Downloading package moses_sample to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping models/moses_sample.zip.
[nltk_data]    | Downloading package movie_reviews to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/movie_reviews.zip.
[nltk_data]    | Downloading package mte_teip5 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/mte_teip5.zip.
[nltk_data]    | Downloading package mwa_ppdb to /root/nltk_data...
[nltk_data]    |   Unzipping misc/mwa_ppdb.zip.
[nltk_data]    | Downloading package names to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/names.zip.
[nltk_data]    | Downloading package nombank.1.0 to /root/nltk_data...
[nltk_data]    | Downloading package nonbreaking_prefixes to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/nonbreaking_prefixes.zip.
[nltk_data]    | Downloading package nps_chat to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/nps_chat.zip.
[nltk_data]    | Downloading package omw to /root/nltk_data...
[nltk_data]    | Downloading package omw-1.4 to /root/nltk_data...
[nltk_data]    | Downloading package opinion_lexicon to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/opinion_lexicon.zip.
[nltk_data]    | Downloading package panlex_swadesh to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    | Downloading package paradigms to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/paradigms.zip.
[nltk_data]    | Downloading package pe08 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/pe08.zip.
[nltk_data]    | Downloading package perluniprops to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping misc/perluniprops.zip.
[nltk_data]    | Downloading package pil to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/pil.zip.
[nltk_data]    | Downloading package pl196x to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/pl196x.zip.
[nltk_data]    | Downloading package porter_test to /root/nltk_data...
[nltk_data]    |   Unzipping stemmers/porter_test.zip.
[nltk_data]    | Downloading package ppattach to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/ppattach.zip.
[nltk_data]    | Downloading package problem_reports to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/problem_reports.zip.
[nltk_data]    | Downloading package product_reviews_1 to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/product_reviews_1.zip.
[nltk_data]    | Downloading package product_reviews_2 to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/product_reviews_2.zip.
[nltk_data]    | Downloading package propbank to /root/nltk_data...
[nltk_data]    | Downloading package pros_cons to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/pros_cons.zip.
[nltk_data]    | Downloading package ptb to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/ptb.zip.
[nltk_data]    | Downloading package punkt to /root/nltk_data...
[nltk_data]    |   Unzipping tokenizers/punkt.zip.
[nltk_data]    | Downloading package qc to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/qc.zip.
[nltk_data]    | Downloading package reuters to /root/nltk_data...
[nltk_data]    | Downloading package rslp to /root/nltk_data...
[nltk_data]    |   Unzipping stemmers/rslp.zip.
[nltk_data]    | Downloading package rte to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/rte.zip.
[nltk_data]    | Downloading package sample_grammars to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping grammars/sample_grammars.zip.
[nltk_data]    | Downloading package semcor to /root/nltk_data...
[nltk_data]    | Downloading package senseval to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/senseval.zip.
[nltk_data]    | Downloading package sentence_polarity to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/sentence_polarity.zip.
[nltk_data]    | Downloading package sentiwordnet to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/sentiwordnet.zip.
[nltk_data]    | Downloading package shakespeare to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/shakespeare.zip.
[nltk_data]    | Downloading package sinica_treebank to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/sinica_treebank.zip.
[nltk_data]    | Downloading package smultron to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/smultron.zip.
[nltk_data]    | Downloading package snowball_data to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    | Downloading package spanish_grammars to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping grammars/spanish_grammars.zip.
[nltk_data]    | Downloading package state_union to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/state_union.zip.
[nltk_data]    | Downloading package stopwords to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/stopwords.zip.
[nltk_data]    | Downloading package subjectivity to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/subjectivity.zip.
[nltk_data]    | Downloading package swadesh to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/swadesh.zip.
[nltk_data]    | Downloading package switchboard to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/switchboard.zip.
[nltk_data]    | Downloading package tagsets to /root/nltk_data...
[nltk_data]    |   Unzipping help/tagsets.zip.
[nltk_data]    | Downloading package timit to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/timit.zip.
[nltk_data]    | Downloading package toolbox to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/toolbox.zip.
[nltk_data]    | Downloading package treebank to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/treebank.zip.
[nltk_data]    | Downloading package twitter_samples to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/twitter_samples.zip.
[nltk_data]    | Downloading package udhr to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/udhr.zip.
[nltk_data]    | Downloading package udhr2 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/udhr2.zip.
[nltk_data]    | Downloading package unicode_samples to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/unicode_samples.zip.
[nltk_data]    | Downloading package universal_tagset to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping taggers/universal_tagset.zip.
[nltk_data]    | Downloading package universal_treebanks_v20 to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    | Downloading package vader_lexicon to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    | Downloading package verbnet to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/verbnet.zip.
[nltk_data]    | Downloading package verbnet3 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/verbnet3.zip.
[nltk_data]    | Downloading package webtext to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/webtext.zip.
[nltk_data]    | Downloading package wmt15_eval to /root/nltk_data...
[nltk_data]    |   Unzipping models/wmt15_eval.zip.
[nltk_data]    | Downloading package word2vec_sample to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping models/word2vec_sample.zip.
[nltk_data]    | Downloading package wordnet to /root/nltk_data...
[nltk_data]    | Downloading package wordnet2021 to /root/nltk_data...
[nltk_data]    | Downloading package wordnet2022 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/wordnet2022.zip.
[nltk_data]    | Downloading package wordnet31 to /root/nltk_data...
[nltk_data]    | Downloading package wordnet_ic to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/wordnet_ic.zip.
[nltk_data]    | Downloading package words to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/words.zip.
[nltk_data]    | Downloading package ycoe to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/ycoe.zip.
[nltk_data]    |
[nltk_data]  Done downloading collection all
```

Out[2]:
```
True
```

In [3]:
```python
text1 = """
... Muad'Dib learned rapidly because his first training was in how to learn.
... And the first lesson of all was the basic trust that he could learn.
... It's shocking to find how many people do not believe they can learn,
... and how many more believe learning to be difficult."""

text2 = """
    Natural-language processing (NLP) is an area of
    computer science and artificial intelligence
    concerned with the interactions between computers
    and human (natural) languages.
"""
```

In [4]:
```python
def extract_ngrams(data, num):
  n_grams = ngrams(word_tokenize(data), num)
  return [ ' '.join(grams) for grams in n_grams]
```

In [5]:
```python
ngrams1 = extract_ngrams(text1,1)
ngrams2 = extract_ngrams(text1,2)
ngrams3 = extract_ngrams(text1,3)
ngrams4 = extract_ngrams(text1,4)
print("ngrams1: ",ngrams1)
print("ngrams2: ",ngrams2)
print("ngrams3: ",ngrams3)
print("ngrams4: ",ngrams4)
```

```
ngrams1:  ["Muad'Dib", 'learned', 'rapidly', 'because', 'his', 'first', 'training', 'was', 'in', 'how', 'to', 'learn', '.', 'And', 'the', 'first', 'lesson', 'of', 'all', 'was', 'the', 'basic', 'trust', 'that', 'he', 'could', 'learn', '.', 'It', "'s", 'shocking', 'to', 'find', 'how', 'many', 'people', 'do', 'not', 'believe', 'they', 'can', 'learn', ',', 'and', 'how', 'many', 'more', 'believe', 'learning', 'to', 'be', 'difficult', '.']
ngrams2:  ["Muad'Dib learned", 'learned rapidly', 'rapidly because', 'because his', 'his first', 'first training', 'training was', 'was in', 'in how', 'how to', 'to learn', 'learn .', '. And', 'And the', 'the first', 'first lesson', 'lesson of', 'of all', 'all was', 'was the', 'the basic', 'basic trust', 'trust that', 'that he', 'he could', 'could learn', 'learn .', '. It', "It 's", "'s shocking", 'shocking to', 'to find', 'find how', 'how many', 'many people', 'people do', 'do not', 'not believe', 'believe they', 'they can', 'can learn', 'learn ,', ', and', 'and how', 'how many', 'many more', 'more believe', 'believe learning', 'learning to', 'to be', 'be difficult', 'difficult .']
ngrams3:  ["Muad'Dib learned rapidly", 'learned rapidly because', 'rapidly because his', 'because his first', 'his first training', 'first training was', 'training was in', 'was in how', 'in how to', 'how to learn', 'to learn .', 'learn . And', '. And the', 'And the first', 'the first lesson', 'first lesson of', 'lesson of all', 'of all was', 'all was the', 'was the basic', 'the basic trust', 'basic trust that', 'trust that he', 'that he could', 'he could learn', 'could learn .', 'learn . It', ". It 's", "It 's shocking", "'s shocking to", 'shocking to find', 'to find how', 'find how many', 'how many people', 'many people do', 'people do not', 'do not believe', 'not believe they', 'believe they can', 'they can learn', 'can learn ,', 'learn , and', ', and how', 'and how many', 'how many more', 'many more believe', 'more believe learning', 'believe learning to', 'learning to be', 'to be difficult', 'be difficult .']
ngrams4:  ["Muad'Dib learned rapidly because", 'learned rapidly because his', 'rapidly because his first', 'because his first training', 'his first training was', 'first training was in', 'training was in how', 'was in how to', 'in how to learn', 'how to learn .', 'to learn . And', 'learn . And the', '. And the first', 'And the first lesson', 'the first lesson of', 'first lesson of all', 'lesson of all was', 'of all was the', 'all was the basic', 'was the basic trust', 'the basic trust that', 'basic trust that he', 'trust that he could', 'that he could learn', 'he could learn .', 'could learn . It', "learn . It 's", ". It 's shocking", "It 's shocking to", "'s shocking to find", 'shocking to find how', 'to find how many', 'find how many people', 'how many people do', 'many people do not', 'people do not believe', 'do not believe they', 'not believe they can', 'believe they can learn', 'they can learn ,', 'can learn , and', 'learn , and how', ', and how many', 'and how many more', 'how many more believe', 'many more believe learning', 'more believe learning to', 'believe learning to be', 'learning to be difficult', 'to be difficult .']
```

> Computing frequency distribution for all the bigrams in the text

In [6]:
```python
def generate_ngrams(s, n):
    # Convert to lowercases
    s = s.strip()
    s = s.lower()
    # Replace all none alphanumeric characters with spaces
    s = re.sub(r'[^a-zA-Z0-9\s]', ' ', s)
    # print(s)

    # Break sentence in the token, remove empty tokens
    tokens = [token.strip() for token in s.split(" ") if token != ""]
    # Use the zip function to help us generate n-grams
    # Concatentate the tokens into ngrams and return
    ngrams = zip(*[tokens[i:] for i in range(n)])
    return [" ".join(ngram) for ngram in ngrams]
```

In [7]:
```python
bigram=generate_ngrams(text2, n=2)
bigram,len(bigram)
```

Out[7]:
```
(['natural language',
  'language processing',
  'processing nlp',
  'nlp is',
  'is an',
  'an area',
  'area of',
  'of computer',
  'computer science',
  'science and',
  'and artificial',
  'artificial intelligence',
  'intelligence concerned',
  'concerned with',
  'with the',
  'the interactions',
  'interactions between',
  'between computers',
  'computers and',
  'and human',
  'human natural',
  'natural languages'],
 22)
```

In [42]:
```python
unigram=generate_ngrams(text2, n=1)
# unigram = unigram.unique()
unigram,len(unigram)
```

Out[42]:
```
(['natural',
  'language',
  'processing',
  'nlp',
  'is',
  'an',
  'area',
  'of',
  'computer',
  'science',
  'and',
  'artificial',
  'intelligence',
  'concerned',
  'with',
  'the',
  'interactions',
  'between',
  'computers',
  'and',
  'human',
  'natural',
  'languages'],
 23)
```

In [44]:
```python
uunigram = []
for i in list(unigram):
  if i not in uunigram:
    uunigram.append(i)
```

In [9]:
```python
clean_txt = ' '.join(i for i in unigram)
clean_txt
```

Out[9]:
```
'natural language processing nlp is an area of computer science and artificial intelligence concerned with the interactions between computers and human natural languages'
```

In [10]:
```python
bgs=extract_ngrams(clean_txt, 2)
fdist = nltk.FreqDist(bgs)
print(fdist)
bigram_lister = []
frequency = []
for k,v in fdist.items():
  # if v > 1:
  print(k,v)
  bigram_lister.append(k.split(' '))
  frequency.append(v)
```

```
<FreqDist with 22 samples and 22 outcomes>
natural language 1
language processing 1
processing nlp 1
nlp is 1
is an 1
an area 1
area of 1
of computer 1
computer science 1
science and 1
and artificial 1
artificial intelligence 1
intelligence concerned 1
concerned with 1
with the 1
the interactions 1
interactions between 1
between computers 1
computers and 1
and human 1
human natural 1
natural languages 1
```

In [11]:
```python
bigram_lister[0]
```

Out[11]:
```
['natural', 'language']
```

In [46]:
```python
n = len(uunigram)
values = [[0 for i in range(n)] for j in range(n)]
```

In [47]:
```python
len(values[0])
```

Out[47]:
```
21
```

In [48]:
```python
df = pd.DataFrame(values,columns = uunigram,index =uunigram)
df
```

Out[48]:

	natural	language	processing	nlp	is	an	area	of	computer	science	...	artificial	intelligence	concerned	with	the	interactions	between	computers	human	languages
natural	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
language	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
processing	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
nlp	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
is	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
an	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
area	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
of	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
computer	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
science	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
and	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
artificial	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
intelligence	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
concerned	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
with	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
the	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
interactions	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
between	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
computers	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
human	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
languages	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

21 rows × 21 columns

In [49]:
```python
for index,value in enumerate(bigram_lister):
  print(value[0],value[1],frequency[index])
  df.loc[value[0],value[1]] = frequency[index]
```

```
natural language 1
language processing 1
processing nlp 1
nlp is 1
is an 1
an area 1
area of 1
of computer 1
computer science 1
science and 1
and artificial 1
artificial intelligence 1
intelligence concerned 1
concerned with 1
with the 1
the interactions 1
interactions between 1
between computers 1
computers and 1
and human 1
human natural 1
natural languages 1
```

In [50]:
```python
df
```

Out[50]:

	natural	language	processing	nlp	is	an	area	of	computer	science	...	artificial	intelligence	concerned	with	the	interactions	between	computers	human	languages
natural	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
language	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
processing	0	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
nlp	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
is	0	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
an	0	0	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0	0	0	0	0
area	0	0	0	0	0	0	0	1	0	0	...	0	0	0	0	0	0	0	0	0	0
of	0	0	0	0	0	0	0	0	1	0	...	0	0	0	0	0	0	0	0	0	0
computer	0	0	0	0	0	0	0	0	0	1	...	0	0	0	0	0	0	0	0	0	0
science	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
and	0	0	0	0	0	0	0	0	0	0	...	1	0	0	0	0	0	0	0	1	0
artificial	0	0	0	0	0	0	0	0	0	0	...	0	1	0	0	0	0	0	0	0	0
intelligence	0	0	0	0	0	0	0	0	0	0	...	0	0	1	0	0	0	0	0	0	0
concerned	0	0	0	0	0	0	0	0	0	0	...	0	0	0	1	0	0	0	0	0	0
with	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	1	0	0	0	0	0
the	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	1	0	0	0	0
interactions	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	0	0	0
between	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	0
computers	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
human	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
languages	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

21 rows × 21 columns

> Finding the probability of each word

In [51]:
```python
df2 = df.copy()
```

In [52]:
```python
df = df2.copy()
```

In [53]:
```python
bgs=extract_ngrams(clean_txt, 1)
fdist = nltk.FreqDist(bgs)
print(fdist)
unigram_lister = []
freq = []
for k,v in fdist.items():
  # if v > 1:
  print(k,v)
  unigram_lister.append(k.split(' '))
  freq.append(v)
```

```
<FreqDist with 21 samples and 23 outcomes>
natural 2
language 1
processing 1
nlp 1
is 1
an 1
area 1
of 1
computer 1
science 1
and 2
artificial 1
intelligence 1
concerned 1
with 1
the 1
interactions 1
between 1
computers 1
human 1
languages 1
```

In [54]:
```python
for index,value in enumerate(unigram_lister):
  print(value,freq[index])
  df.loc[value] = df.loc[value]/freq[index]
```

```
['natural'] 2
['language'] 1
['processing'] 1
['nlp'] 1
['is'] 1
['an'] 1
['area'] 1
['of'] 1
['computer'] 1
['science'] 1
['and'] 2
['artificial'] 1
['intelligence'] 1
['concerned'] 1
['with'] 1
['the'] 1
['interactions'] 1
['between'] 1
['computers'] 1
['human'] 1
['languages'] 1
```

In [55]:
```python
list(df.iloc[0].index)
```

Out[55]:
```
['natural',
 'language',
 'processing',
 'nlp',
 'is',
 'an',
 'area',
 'of',
 'computer',
 'science',
 'and',
 'artificial',
 'intelligence',
 'concerned',
 'with',
 'the',
 'interactions',
 'between',
 'computers',
 'human',
 'languages']
```

In [56]:
```python
df['natural']
```

Out[56]:
```
natural         0.0
language        0.0
processing      0.0
nlp             0.0
is              0.0
an              0.0
area            0.0
of              0.0
computer        0.0
science         0.0
and             0.0
artificial      0.0
intelligence    0.0
concerned       0.0
with            0.0
the             0.0
interactions    0.0
between         0.0
computers       0.0
human           1.0
languages       0.0
Name: natural, dtype: float64
```

In [59]:
```python
df.head()
```

Out[59]:

	natural	language	processing	nlp	is	an	area	of	computer	science	...	artificial	intelligence	concerned	with	the	interactions	between	computers	human	languages
natural	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5
language	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
processing	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
nlp	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
is	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 21 columns

In [60]:
```python
def predict_nextword(word,df):
  columns = list(df.columns)
  word_columns = list(df[word].index)
  max_value = max(df[word])
  index = 0
  if word in columns:
    for i in df[word]:
      index += 1
      if max_value == i:
        return word_columns[index]

  else:
    print("Enter the bigram word from the word corpora")

word = "natural"
predict_nextword(word,df)
```

Out[60]:
```
'languages'
```

In [ ]: