ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING – SITE

Department of MSC in Artificial Intelligence

Course Assignment

> Course: NLP
>
> Name: Mintesnot Fikir
>
> Section: Regular
>
> IDs: GSR1669/15
>
> Submitted to: Fantahun B. (PhD)
>
> Submission date November 2023

NLP Assignment-1 (12%)

Fantahun Bogale • Oct 20 (Edited Nov 6) 12 points Due Yesterday, 11:59 PM

- This assignment has three (3) tasks each having its own weight. The total weight of this assignment will be 12%.
- This assignment is based on one version of the Amharic Corpus (GPAC) created by this paper. https://doi.org/10.3390/info12010020.
- You can access this corpus using this link: link. https://drive.google.com/file/d/1jEOAORyITBAiNKes_JerD6gFF7cnLFyr/view?usp=sharing
- This assignment is an individual assignment. Experience sharing and discussion is possible, however each student is responsible to submit and defend his work.
- Presentation may be required (get ready for that) >

Note: The corpus may require a good size of RAM to load; hence you can make use of Google Collab, Kaggle or more if your device is incapable. > Questions:

1 N-gram language model (6%)

- 1.1 Create n-grams for n=1, 2, 3, 4. You can show sample prints.
- 1.2 Calculate probabilities of n-grams and find the top 10 most likely n-grams for all n.
- 1.3 What is the probability of the sentence. "ኢትዮጵያ ታሪካዊ ሀገር ናት ". You can also try more sentences.
- 1.4 Generate random sentences using n-grams; explain what happens as n-increases based on your output. >

2 Evaluate these Language Models Using Intrinsic Evaluation Method (3%)

3 Evaluate these Language Models Using Extrinsic Evaluation Method (3%)

```
    You can make use of any task convenient for you to evaluate the
n-gram models you have created.
```

Solution

Outline

Preprocessing

Import necessary libraries

In [1]:
```
import pickle
import nltk
import random
import gc
from nltk import ngrams
from nltk.util import ngrams
from nltk import FreqDist
from collections import Counter
import random
import math
import time
import string
```

Reading the corpus from local folder(text file) as a text file

In [2]:
```
starttime = time.time()
with open('/kaggle/input/amharic-corpus-general/GPAC.txt', 'r', encoding='utf-8') a
    text = file.read()
endtime = time.time()
print(f"Time taken for read the text as a string is :  {(endtime - starttime)/60}
```

```
Time taken for read the text as a string is :  0.3450124700864156  minuets
```

In [3]:
```
print(type(text))
```

```
<class 'str'>
```

As we have imported text file as a single string, next step is to make it comfortable for farther processing, so we need to tokenize the string into list of tokens

In [4]:
```
print("\n\n Sample text data:\n\n",text[4:3000])
```

```
 Sample text data:

 ምን መሰላችሁ? (አንባቢያን) ኢትዮጵያ በተደጋጋሚ ጥሪው ደርሷት ልትታደመው ያልቻለችው የአለም የእግር ኳስ ዋ ለ19ኛ ጊዜ በደቡብ አፍሪካ ሲጠጣ፣ በሩቅ እያዮች አንጀቷ ባረረ ልክ በአመቱ በለስ ቀናትና ሌላ ዋ ልትታደም ሁለት ልጆቿን ወደ ደቡብ አፍ ሪካ ላከች::6ኛው ቢግ ብራዘርስ አፍሪካ አብሮ የመኖር ውድድር በደቡብ አፍሪካ ተካሂዷል:: ከተለያዩ 14 የአፍሪካ አገራት የተው ጣጡ 26 ያህል ተሳታፊዎች የተካፈሉበት ይህ ውድድር፣ ግለሰቦች በፈታኝ ሁኔታ ውስጥ በማለፍ ብቃታቸውን የሚያስመሰክሩበት መሆኑን ሰምተናል:: የሚገጥሟቸውን የተለያዩ ፈተናዎች በትእግስትና በጥበብ ማለፍ፣ ከሌሎች ጋር ተስማምቶ መዝለቅ፣ ችግሮችን በ ብልጠት መፍታት ወዘተ በየጊዜው ከሚደረገው ቅነሳ ተርፈው ለ91 ቀናት ያህል በውድድሩ መቆየት የቻሉ ሁለት ተወዳዳሪዎ ች እያንዳንዳቸው 200 ሺህ ዶላር እንደሚሸለሙም ሰናገር ነበር:: በዘንድሮው ውድድር አገራችን ዳኒ እና ሃኒ የተባሉ ሁለት ወጣ ቶችን ብታሰልፍም ዳኒ ቀደም ብሎ የቅነሳው ሰለባ ሲሆን ሃኒም በቅርቡ ከውድድር ውጭ ሆናለች::ይህቺን የአገሪቱ ብቸኛ ተስፋ ወ ደ አሸናፊነት ለማሸጋገር የህዝብ የድጋፍ ድም ወሳኝ መሆኑን የተገነዘበው ወዳጄ ነው እንግዲህ 835 የሚ ል አገራዊ ጥሪ ያስተላለፈልኝ ያኔ ሃኒ ከውድድሩ ከመሰናበቷ በፊት::ወዳጄ የአገሩን ስም በአሸናፊነት የማስጠራት ከፍተኛ ጉጉት፣ አገሬ እንዳትሸነፍ የሚል ከፍተኛ ስጋት እንዳደረበት ይሰማኛል:: ጉጉቱ ሳይሆን ስጋቱ የወዳጄን የዋህነት ::ሃኒም ኢትዮ ጵያም ይሸነፉ ይሆን? በሚል እንዲህ ከንቱ ስጋት የሚያንገበግባቸውን አገር ወዳድ ዜጐች እኔ የዋሆች እላቸዋለሁ::የዋሆች ሆይ!አት ስጉ ስለ ሃኒም ስለ ኢትዮጵያም አትስጉ:: ውድድሩ ቢግ ብራዘርስ አፍሪካ በአርቴፊሻል ፈተናዎች ውስጥ አልፎ ለሦስት ወ ራት የመቆየት ውድድር ነው:: ደቡብ አፍሪካ ለስድስተኛ ጊዜ ካዘጋጀችው ቢግ ብራዘርስ የመረረ ውድድር ለሺህ አመታት በተከታታ ይ ስታዘጋጅ የኖረች፣ መላ ህዝቧን አሳትፋ መላ ህዝቧን ስትሸልም የኖረች አገር ናት ኢትዮጵያ!የደቡብ አፍሪካው እንጂ የኢትዮ ጵያው ቢግ ብራዘርስ ሶስት ወር ተብሎ ቀን የሚቆረጥለት ውስን የፈተና ጊዜ የለውም:: አንዲት ኢትዮጵያ በአንዲት ሃኒ ሳይሆን በ መላው ህዝቧ ነው የምትወከለው ለምን ቢባል ሶስት ወር ሳይሆን ሶስት ሺህ ዘመን በራሷ ቢግ ብራዘርስ ተካፍላ የምንጊዜም አሸናፊ ሆና ዘልቃለችና::ኢትዮጵያ ለዘመናት ባስተናገደችው የራሷ ቢግ ብራዘርስ አቻ የለሽ ፈተና ውስጥ መላው ህዝቧን እያሳተፈች ድሉን ከህዝቧ እጅ ባለማስነጠቅ ሃትሪክ የሰራች (በሺህ አመታት ስሌት) የምንጊዜም ድል ባለቤት እኮ ናት!የዋሆች ስለ ደቡብ አፍሪካው ቢግ ብራዘር ስለምን ትጨነቃላችሁ? ሃኒ እኮ ለእግር ኳስ አይደለም የሄደችው:: እሱንማ ብለነው ብለነው አልሆን ብ ሎ ቸግሮናል:: ደረጃችን ከሌሎች በታች ሆኖ ቀርቶብን በሩቅ እያየነው ተብሰልስለናል:: አሁን ሃኒን ወደ ደቡብ አፍሪካ የላክናት ከአ ቅሟ በላይ ሳይሆን በታች ወርዳ ወደምትጫወትበት አብሮ የመኖር ቀላል ፉክክር ነው:: ሰቆቃንና ፈተናን ተጋፍጦ በመኖር የአለም ሻምፒዮናነቱን አለም በአንድ ድም ያፀደቀለት አሸናፊ ህዝብ ወኪል የሆነችው ሃኒ፣ ያለ ዲቪዚዮኗ ስንትና ስንት ቁልቁል ወርዳ እኮ ነ ው የተወዳደረችው::ሃኒ ከቢግ ብራዘር ውድድር ውጭ መሆኗን ሰምኑን ሰማሁ:: ሰማሁና ሳቅኩ:: ለምን ሳቅኩ? የአገሬ መሸነፍ የ ማያንገበግበኝ ሰው ሆኜ ነውን? አይመስለኝም!ሃኒ ከውድድሩ ውጭ የሆነችው የቢግ ብራዘርስ አብሮ የመኖር ውድድር አሸንፏ ት ወይም አቅቷት አይመስለኝም! እሷ ከመስፈርቱ በላይ ሆና እንጂ! ( እንደ ለት) ኢትዮጵያ ከ6ኛ ው የቢግ ብራዘርስ አፍሪካ ውድድር የተሰናበተችው በቀላል ሚዛን ውድድር የከባድ ሚዛን ተወዳዳሪ በማሰለፏ ነው ባይ ነኝ:: ልክ በእግር ኳስ ውድድር ላይ እንደሚከሰተው ለምሳሌ ከ16 አመት በታች የሆኑ ተጫዋቾች በሚሳተፉበት የታዳጊ ወጣቶች ሻምፒ ዮና ላይ የ25 አመት እድሜ ያለው ተጫዋች በማሰለፉ ከውድድር ውጭ እንደሚሆን ከሉ::ይመስለኛል ሃኒ ከውድድሩ የተባረ ረችው የቢግ ብራዘርስ ፈተና ስላቃታት አይደለም:: ምናልባትም እሷ ለፈተናው ከብዳው ቢሆን እንጂ:: ሃኒ ከቢግ ብራዘር ግቢ ተ ባረረች የሚሉ፣ እነሱ አንዷን ሃኒ ብቻ የሚያዩ የዋሆች ናቸው:: ሃኒ እዚያው ደቡብ አፍሪካ ናት ደቡብ አፍሪካ ኡጋንዳ ኬኒያ ሊቢያ ሻሸመኔ ቤሩት አሜሪካ እዚህም እዚያም ተበትና ከቢግ ብራዘር የመረረ ሰቆቃ ውስጥ እየተንገላ ታች ችግር ቻይነቷን እያስመሰከረች ያለች ብዙ ኢትዮጵያዊ ሴት ናት ሃኒ::ቢግ ብራዘር በሚሉት የፌክ ፈተና እና ፎርጅድ ውጣ ው ረድ ተሸነፋችሁ ሲሉን ሰማሁና ሳቅኩ!ዳኒ እና ሃኒ አይችሉም ተብለው መባረራቸው በሳቅ አፈረሰኝ::ምን ማለታቸው ነው ዳኞቹ? እንደ ዳኒ እንደ ሃኒ እንደ መላው ኢትዮጵያዊ ለችግር ሳይረታ ዘመናትን ያለፈ ማን ነው? ተቻችሎ መኖር ከሆነ ጉዳ ዩ ማን እንደነሱ ተቻቻይ አለና ነው! ስድብ ዘለፋን አይደለም፣ ግርፋትን ችሎ የኖረ ቆዳው ድርብ ህዝብ እኮ ነው ተሸን ፈሏል የተባለው::ከዚህ በላይ ኮሜዲ አለ እንዴ?ለሶስት ሺህ ዘመን ክፉ ደጉን ችሎ አብሮ የኖረ ህዝብ እንዴት ነው ለሶስት ወር አ ብሮ መኖር አቃተህ ተብሎ ቀይ ካርድ የሚሰጠው?ሃኒ እና ዳኒ እኮ በቢ
```

The dataset contains numerous extraneous characters and symbols which require further processing. Below is a list of some of these:

- Ethiopian punctuation marks: '፡', '።', '፣', '፤', '፥', '፦', '፧', '፨';
- Lowercase letters: 'a' to 'z';
- Uppercase letters: 'A' to 'Z';
- Numerals: '0' to '9';
- Special characters and symbols: '!', '"', '#', '$', '%', '&', "'", '(', ')', '*', '+', ',', '-', '.', '/', ':', ';', '<', '=', '>', '?', '@', '[', '\', ']', '^', '_', '`', '{', '|', '}', '~'.

Tokenize the corpus

To facilitate more efficient processing of the corpus, the initial phase should involve tokenization. In this case, since the text comprises a mix of individual characters, symbols, and potential punctuation marks, tokenization would involve categorizing these elements into meaningful groups that can be separately identified and analyzed.

By breaking down the text into tokens, we can apply subsequent processing steps more effectively. These steps could include filtering out unnecessary symbols, converting characters to a uniform case for consistency, or identifying and preserving meaningful punctuation for further linguistic analysis. Tokenization simplifies these tasks by providing a structured approach to dissecting the text, allowing for a more manageable and targeted processing workflow.

we have 2 ways of tokenization either create Custom Tokenization Function from scrach or use third party library

In [5]:
```python
# tokenizer functions
import string
def tokenizer(text: str):
    tokens = []
    current_token = ''

    for char in text:
        if char.isspace() or char in string.punctuation:
            if current_token:
                tokens.append(current_token)
                current_token = ''
        else:
            current_token += char

    if current_token:
        tokens.append(current_token)

    return tokens
```

- The above custom tokenization function tokenizer iterates over each character in the input string. It builds tokens by appending characters until it encounters a whitespace or a punctuation character, at which point it adds the current token to the list and starts building the next one. This function does not use regular expressions and relies purely on sequential character checking, making it straightforward but potentially less efficient for large texts compared to regex-based solutions.

In [6]:
```python
start_time = time.time()
tokens = tokenizer(text)
```

```
end_time = time.time()
print(f"\n\nTime taken for tokenizetion:  {(end_time - start_time)/60}  minutes \n'
print(tokens[:200])
print('\nSize of valid tokens are: ',len(tokens),'Words(tokens)')
```

```
Time taken for tokenizetion:  2.299083085854848  minutes
```

```
['ምን', 'መሰላችሁ', 'አንባቢያን', 'ኢትዮጵያ', 'በተደጋጋሚ', 'ጥሪው', 'ደርሷት', 'ልትታደመው', 'ያልቻለች
ው', 'የአለም', 'የእግር', 'ኳስ', 'ዋ', 'ለ19ኛ', 'ጊዜ', 'በደቡብ', 'አፍሪካ', 'ሲጠጣ፣', 'በሩቅ', 'እያየ
ች', 'አንጀቷ', 'ባረረ', 'ልክ', 'በአመቱ', 'በለስ', 'ቀናትና', 'ሌላ', 'ዋ', 'ልትታደም', 'ሁለት', 'ልጆቿ
ን', 'ወደ', 'ደቡብ', 'አፍሪካ', 'ላኩች።6ኛው', 'ቢግ', 'ብራዘርስ', 'አፍሪካ', 'አብሮ', 'የመኖር', 'ውድድ
ር', 'በደቡብ', 'አፍሪካ', 'ተካሂዷል።', 'ከተለያዩ', '14', 'የአፍሪካ', 'አገራት', 'የተውጣጡ', '26', 'ያህ
ል', 'ተሳታፊዎች', 'የተካፈሉበት', 'ይህ', 'ውድድር፣', 'ግለሰቦች', 'በፈታኝ', 'ሁኔታ', 'ውስጥ', 'በማለፍ',
'ብቃታቸውን', 'የሚያስመሰክሩበት', 'መሆኑን', 'ሰምተናል።', 'የሚገጥሟቸውን', 'የተለያዩ', 'ፈተናዎች', 'በትእግስ
ትና', 'በጥበብ', 'ማለፍ፣', 'ከሌሎች', 'ጋር', 'ተስማምቶ', 'መዝለቅ፣', 'ችግሮችን', 'በብልጠት', 'መፍታት',
'ወዘተ', 'በየጊዜው', 'ከሚደረገው', 'ቅነሳ', 'ተርፈው', 'ለ91', 'ቀናት', 'ያህል', 'በውድድሩ', 'መቆየት',
'የቻሉ', 'ሁለት', 'ተወዳዳሪዎች', 'እያንዳንዳቸው', '200', 'ሺህ', 'ዶላር', 'እንደሚሸለሙም', 'ሲናገር', 'ነበ
ር።', 'በዘንድሮው', 'ውድድር', 'አገራችን', 'ዳኒ', 'እና', 'ሃኒ', 'የተባሉ', 'ሁለት', 'ወጣቶችን', 'ብታሰልፍ
ም', 'ዳኒ', 'ቀደም', 'ብሎ', 'የቅነሳው', 'ሰለባ', 'ሲሆን', 'ሃኒም', 'በቅርቡ', 'ከውድድር', 'ውጭ', 'ሆና
ለች።ይህቺን', 'የአገሪቱ', 'ብቸኛ', 'ተስፋ', 'ወደ', 'አሸናፊነት', 'ለማሸጋገር', 'የህዝብ', 'የድጋፍ', 'ድም',
'ወሳኝ', 'መሆኑን', 'የተገነዘበው', 'ወዳጇ', 'ነው', 'እንግዲህ', '835', 'የሚል', 'አገራዊ', 'ጥሪ', 'ያስተላለ
ፈልኝ', 'ያኔ', 'ሃኒ', 'ከውድድሩ', 'ከመሰናበቷ', 'በፊት።ወዳጇ', 'የአገሩን', 'ስም', 'በአሸናፊነት', 'የማስጠራ
ት', 'ከፍተኛ', 'ጉጉት፣', 'አገሬ', 'እንዳትሸነፍ', 'የሚል', 'ከፍተኛ', 'ስጋት', 'እንዳደረበት', 'ይሰማኛል።',
'ጉጉቱ', 'ሳይሆን', 'ስጋቱ', 'የወዳጇን', 'የዋህነት', '።ሃኒም', 'ኢትዮጵያም', 'ይሸነፉ', 'ይሆን', 'በሚል',
'እንዲህ', 'ከንቱ', 'ስጋት', 'የሚያንገበግባቸውን', 'አገር', 'ወዳድ', 'ዜጐች', 'እኔ', 'የዋሆች', 'እላቸዋለሁ።የዋ
ሆች', 'ሆይ', 'አትስጉ', 'ስለ', 'ሃኒም', 'ስለ', 'ኢትዮጵያም', 'አትስጉ።', 'ውድድሩ', 'ቢግ', 'ብራዘርስ',
'አፍሪካ', 'በአርቴፊሻል', 'ፈተናዎች', 'ውስጥ', 'አልፎ', 'ለሦስት', 'ወራት', 'የመቆየት', 'ውድድር', 'ነው፡
፡', 'ደቡብ', 'አፍሪካ', 'ለስድስተኛ', 'ጊዜ']
```

```
Size of valid tokens are:  80769388 Words(tokens)
```

In [7]:
```
start_time = time.time()
tokens1 = nltk.word_tokenize(text)
end_time = time.time()
print(f"Time taken for tokenizetion:  {(end_time - start_time)/60}  minutes")
```

```
Time taken for tokenizetion:  12.268705590565999  minutes
```

Here, the entire file's content is read into one string, and then NLTK's word_tokenize function is used to tokenize the string into words. This method is straightforward and leverages the robust tokenization capabilities of the NLTK library. Now we have list of tokens in a single list variable `tokens1` but it is not `Efficient` so we use tokenization function tokenizer where built from scratch , because it tooks `12.22` minutes, almost 6 times customized one which is `2.24` minutes.

In [8]:
```
print('Token size ',len(tokens))
```

```
Token size  80769388
```

In [9]:
```
print('\n\n Sample  from the tokens: first 500 words are :\n')
print(tokens[:500])
```

```
Sample  from the tokens: first 500 words are :

['ምን', 'መሰላችሁ', 'አንባቢያን', 'ኢትዮጵያ', 'በተደጋጋሚ', 'ጥሪው', 'ደርሷት', 'ልትታደመው', 'ያልቻለች
ው', 'የአለም', 'የእግር', 'ኳስ', 'ዋ', 'ለ19ኛ', 'ጊዜ', 'በደቡብ', 'አፍሪካ', 'ሲጠጣ፣', 'በሩቅ', 'እያየ
ች', 'አንጀቷ', 'ባረረ', 'ልክ', 'በአመቱ', 'በለስ', 'ቀናትና', 'ሌላ', 'ዋ', 'ልትታደም', 'ሁለት', 'ልጆቿ
ን', 'ወደ', 'ደቡብ', 'አፍሪካ', 'ላኩች።6ኛው', 'ቢግ', 'ብራዘርስ', 'አፍሪካ', 'አብሮ', 'የመኖር', 'ውድድ
ር', 'በደቡብ', 'አፍሪካ', 'ተካሂዷል።', 'ከተለያዩ', '14', 'የአፍሪካ', 'አገራት', 'የተውጣጡ', '26', 'ያህ
ል', 'ተሳታፊዎች', 'የተካፈሉበት', 'ይህ', 'ውድድር፣', 'ግለሰቦች', 'በፈታኝ', 'ሁኔታ', 'ውስጥ', 'በማለፍ',
'ብቃታቸውን', 'የሚያስመሰክሩበት', 'መሆኑን', 'ሰምተናል።', 'የሚገጥሟቸውን', 'የተለያዩ', 'ፈተናዎች', 'በትእግስ
ትና', 'በጥበብ', 'ማለፍ፣', 'ከሌሎች', 'ጋር', 'ተስማምቶ', 'መዝለቅ፣', 'ችግሮችን', 'በብልጠት', 'መፍታት',
'ወዘተ', 'በየጊዜው', 'ከሚደረገው', 'ቅነሳ', 'ተርፈው', 'ለ91', 'ቀናት', 'ያህል', 'በውድድሩ', 'መቆየት',
'የቻሉ', 'ሁለት', 'ተወዳዳሪዎች', 'እያንዳንዳቸው', '200', 'ሺህ', 'ዶላር', 'እንደሚሸለሙም', 'ሲናገር', 'ነበ
ር።', 'በዘንድሮው', 'ውድድር', 'አገራችን', 'ዳኒ', 'እና', 'ሃኒ', 'የተባሉ', 'ሁለት', 'ወጣቶችን', 'ብታሰልፍ
ም', 'ዳኒ', 'ቀደም', 'ብሎ', 'የቅነሳው', 'ሰለባ', 'ሲሆን', 'ሃኒም', 'በቅርቡ', 'ከውድድር', 'ውጭ', 'ሆና
ለች።ይህቺን', 'የአገሪቱ', 'ብቸኛ', 'ተስፋ', 'ወደ', 'አሸናፊነት', 'ለማሸጋገር', 'የህዝብ', 'የድጋፍ', 'ድም',
'ወሳኝ', 'መሆኑን', 'የተገነዘበው', 'ወዳጇ', 'ነው', 'እንግዲህ', '835', 'የሚል', 'አገራዊ', 'ጥሪ', 'ያስተላለ
ፈልኝ', 'ያኔ', 'ሃኒ', 'ከውድድሩ', 'ከመሰናበቷ', 'በፊት።ወዳጇ', 'የአገሩን', 'ስም', 'በአሸናፊነት', 'የማስጠራ
ት', 'ከፍተኛ', 'ጉጉት፣', 'አገሬ', 'እንዳትሸነፍ', 'የሚል', 'ከፍተኛ', 'ስጋት', 'እንዳደረበት', 'ይሰማኛል።',
'ጉጉቱ', 'ሳይሆን', 'ስጋቱ', 'የወዳጇን', 'የዋህነት', '።ሃኒም', 'ኢትዮጵያም', 'ይሸነፉ', 'ይሆን', 'በሚል',
'እንዲህ', 'ከንቱ', 'ስጋት', 'የሚያንገበግባቸውን', 'አገር', 'ወዳድ', 'ዜጐች', 'እኔ', 'የዋሆች', 'እላቸዋለሁ።የዋ
ሆች', 'ሆይ', 'አትስጉ', 'ስለ', 'ሃኒም', 'ስለ', 'ኢትዮጵያም', 'አትስጉ።', 'ውድድሩ', 'ቢግ', 'ብራዘርስ',
'አፍሪካ', 'በአርቴፊሻል', 'ፈተናዎች', 'ውስጥ', 'አልፎ', 'ለሥስት', 'ወራት', 'የመቆየት', 'ውድድር', 'ነው፡
፡', 'ደቡብ', 'አፍሪካ', 'ለስድስተኛ', 'ጊዜ', 'ካዘጋጀችው', 'ቢግ', 'ብራዘርስ', 'የመረረ', 'ውድድር', 'ለሺ
ህ', 'አመታት', 'በተከታታይ', 'ስታዘጋጅ', 'የኖረች፣', 'መላ', 'ህዝቧን', 'አሳትፋ', 'መላ', 'ህዝቧን', 'ስትሸ
ልም', 'የኖረች', 'አገር', 'ናት', 'ኢትዮጵያ', 'የደቡብ', 'አፍሪካው', 'እንጂ', 'የኢትዮጵያው', 'ቢግ', 'ብራዘ
ርስ', 'ሶስት', 'ወር', 'ተብሎ', 'ቀን', 'የሚቆረጥለት', 'ውስን', 'የፈተና', 'ጊዜ', 'የለውም።', 'አንዲት',
'ኢትዮጵያ', 'በአንዲት', 'ሃኒ', 'ሳይሆን', 'በመላው', 'ህዝቧ', 'ነው', 'የምትወከለው', 'ለምን', 'ቢባል',
'ሶስት', 'ወር', 'ሳይሆን', 'ሶስት', 'ሺህ', 'ዘመን', 'በራሷ', 'ቢግ', 'ብራዘርስ', 'ተካፍላ', 'የምንጊዜም',
'አሸናፊ', 'ሆና', 'ዘልቃለችና።ኢትዮጵያ', 'ለዘመናት', 'ባስተናገደችው', 'የራሷ', 'ቢግ', 'ብራዘርስ', 'አቻ', 'የ
ለሽ', 'ፈተና', 'ውስጥ', 'መላው', 'ህዝቧን', 'እያሳተፈች', 'ድሉን', 'ከህዝቧ', 'እጅ', 'ባለማስነጠቅ', 'ሃትሪ
ክ', 'የሰራች', 'በሺህ', 'አመታት', 'ስሌት', 'የምንጊዜም', 'ድል', 'ባለቤት', 'እኮ', 'ናት', 'የዋሆች', 'ስ
ለ', 'ደቡብ', 'አፍሪካው', 'ቢግ', 'ብራዘር', 'ስለምን', 'ትጨነቃላችሁ', 'ሃኒ', 'እኮ', 'ለእግር', 'ኳስ',
'አይደለም', 'የሄደችው።', 'እሱንማ', 'ብለነው', 'ብለነው', 'አልሆን', 'ብሎ', 'ቸግሮናል።', 'ደረጃችን', 'ከሌ
ሎች', 'በታች', 'ሆኖ', 'ቀርቶብን', 'በሩቅ', 'እያየነው', 'ተብሰልስለናል።', 'አሁን', 'ሃኒን', 'ወደ', 'ደቡ
ብ', 'አፍሪካ', 'የላከናት', 'ከአቅሟ', 'በላይ', 'ሳይሆን', 'በታች', 'ወርዳ', 'ወደምትጫወትበት', 'አብሮ', 'የ
መኖር', 'ቀላል', 'ፉክክር', 'ነው።', 'ሰቆቃንና', 'ፈተናን', 'ተጋፍጦ', 'በመኖር', 'የአለም', 'ሻምፒዮናነቱን',
'አለም', 'በአንድ', 'ድም', 'ያፀደቀለት', 'አሸናፊ', 'ህዝብ', 'ወኪል', 'የሆነችው', 'ሃኒ፣', 'ያለ', 'ዲቪዚዮ
ኗ', 'ስንትና', 'ስንት', 'ቁልቁል', 'ወርዳ', 'እኮ', 'ነው', 'የተወዳደረችው።ሃኒ', 'ከቢግ', 'ብራዘር', 'ውድድ
ር', 'ውጭ', 'መሆኗን', 'ሰሞኑን', 'ሰማሁ።', 'ሰማሁና', 'ሳቅኩ።', 'ለምን', 'ሳቅኩ', 'የአገሬ', 'መሸነፍ',
'የማያንገበግበኝ', 'ሰው', 'ሆኜ', 'ነውን', 'አይመስለኝም', 'ሃኒ', 'ከውድድሩ', 'ውጭ', 'የሆነችው', 'የቢግ',
'ብራዘርስ', 'አብሮ', 'የመኖር', 'ውድድር', 'አሸንፏት', 'ወይም', 'አቅቷት', 'አይመስለኝም', 'እሷ', 'ከመስፈር
ቱ', 'በላይ', 'ሆና', 'እንጂ', 'እንደ', 'ሉት', 'ኢትዮጵያ', 'ከ6ኛው', 'የቢግ', 'ብራዘርስ', 'አፍሪካ', 'ውድ
ድር', 'የተሰናበተችው', 'በቀላል', 'ሚዛን', 'ውድድር', 'የከባድ', 'ሚዛን', 'ተወዳዳሪ', 'በማሰለፏ', 'ነው',
'ባይ', 'ነኝ።', 'ልክ', 'በእግር', 'ኳስ', 'ውድድር', 'ላይ', 'እንደሚከሰተው', 'ለምሳሌ', 'ከ16', 'አመት',
'በታች', 'የሆኑ', 'ተጫዋቾች', 'በሚሰተፉበት', 'የታዳጊ', 'ወጣቶች', 'ሻምፒዮና', 'ላይ', 'የ25', 'አመት',
'እድሜ', 'ያለው', 'ተጫዋች', 'በማሰለፉ', 'ከውድድር', 'ውጭ', 'እንደሚሆን', 'ክለብ።ይመስለኛል', 'ሃኒ',
'ከውድድሩ', 'የተባረረችው', 'የቢግ', 'ብራዘርስ', 'ፈተና', 'ስላቃታት', 'አይደለም።', 'ምናልባትም', 'እሷ',
'ለፈተናው', 'ከብዳው', 'ቢሆን', 'እንጂ።', 'ሃኒ', 'ከቢግ', 'ብራዘር', 'ግቢ', 'ተባረረች', 'የሚሉ፣', 'እነ
ሱ', 'አንዷን', 'ሃኒ', 'ብቻ', 'የሚያዩ', 'የዋሆች', 'ናቸው።', 'ሃኒ', 'እዚያው', 'ደቡብ', 'አፍሪካ', 'ና
ት', 'ደቡብ', 'አፍሪካ', 'ኡጋንዳ', 'ኬኒያ', 'ሊቢያ', 'ሻሸመኔ', 'ቤሩት', 'አሜሪካ', 'እዚህም', 'እዚያም',
'ተበትና', 'ከቢግ', 'ብራዘር', 'የመረረ', 'ሰቆቃ', 'ውስጥ', 'እየተንባላታች', 'ችግር', 'ቻይነቷን', 'እያስመሰከረ
ች', 'ያለች', 'ብዙ', 'ኢትዮጵያዊ', 'ሴት', 'ናት', 'ሃኒ።ቢግ', 'ብራዘር', 'በሚሉት', 'የፌክ', 'ፈተና', 'እ
ና']
```

Save the tokens for later use and change the address to none temporary directory or download and manualy upload to fixed directory

The Kaggle working directory is temporary, and it deletes files upon restarting. To ensure the preservation of tokens for later use, one option is to save them and change the directory to a non-temporary location. Alternatively, the tokens can be downloaded and manually

uploaded to a fixed directory. In my case, I will manually upload the saved tokens to '/kaggle/input/saved-amharic-tokens'.

In [10]:
```
# The Kaggle working directory is temporary, and it deletes files upon restarting.
# To ensure the preservation of tokens for later use, one option is to save them ar
with open('/kaggle/working/tokens.pkl', 'wb') as file:
    pickle.dump(tokens, file)
```

Load saved tokens we have to load the token from different directory

In [11]:
```
# # Load saved tokens we have to load the token from different directory
with open('/kaggle/input/saved-tokens/tokens.pkl', 'rb') as file:
    tokens = pickle.load(file)
```

Clean the corpus from amharic stopwords, English symbols, Numbers, amharic punctuations

After tokenization, the next step is to clean the corpus by removing stopwords, symbols, and punctuation from both the Amharic and English languages.

In [13]:
```
# A sample list of custom stopwords in Amharic
import string

# Define a list of custom stopwords in Amharic
amharic_stopwords = []

# Define Amharic punctuations
amharic_punctuations = ['።', '፡', '፣', '፤', '፥', '፦', '፧', '፨']

# English symbols that we want to remove (this includes letters, digits, and punctu
english_symbols = list(string.ascii_letters + string.digits + string.punctuation)

# Combined list of characters to remove
elemnts_to_remove = amharic_stopwords + amharic_punctuations + english_symbols
start_time = time.time()
cleaned_tokens = [token for token in tokens if not any(ch in elemnts_to_remove for

print('\nTokens removed \n',elemnts_to_remove)

end_time = time.time()
print(f"\n\nTime taken for cleaning  :  {(end_time - start_time)/60}  minutes \n")
```

```
Tokens removed
 ['።', '፡', '፣', '፤', '፥', '፦', '፧', '፨', 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h',
'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x',
'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N',
'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', '0', '1', '2', '3',
'4', '5', '6', '7', '8', '9', '!', '"', '#', '$', '%', '&', "'", '(', ')', '*',
'+', ',', '-', '.', '/', ':', ';', '<', '=', '>', '?', '@', '[', '\\', ']', '^',
'_', '`', '{', '|', '}', '~']


Time taken for cleaning  :  9.963485936323801  minutes
```

First we define the unwanted elements, including Amharic stopwords, Amharic punctuation, and English symbols (comprising letters, digits, and punctuation). The then iterates through

the list of tokens and excludes any token that contains these elements or is composed solely of digits. The resulting cleaned_tokens list contains the text data cleansed of irrelevant characters and symbols, ready for further text analysis or processing tasks.

It then cleans the tokenized data by removing any tokens that contain these defined characters or are purely numeric. The result is a list of cleaned tokens free from unwanted characters.

In [14]:
```
print('\nToken size after cleaned: ', len(cleaned_tokens))
print('\nThe first 50 words from the tokens \n',cleaned_tokens[:150])

print('\nToken size befor cleaned: ',len(tokens))
print('\nThe first 50 words from the original tokens \n',tokens[:150])



print('\nLength difference/Number of removed tokens: ',len(tokens)-len(cleaned_toke
```

```
Token size after cleaned:  71813265

The first 50 words from the tokens
 ['ምን', 'መሰላችሁ', 'አንባቢያን', 'ኢትዮጵያ', 'በተደጋጋሚ', 'ጥሪው', 'ደርሷት', 'ልትታደመው', 'ያልቻሉች
ው', 'የአለም', 'የእግር', 'ኳስ', 'ዋ', 'ጊዜ', 'በደቡብ', 'አፍሪካ', 'በሩቅ', 'እያየች', 'አንጀቷ', 'ባረረ',
'ልክ', 'በአመቱ', 'በለስ', 'ቀናትና', 'ሌላ', 'ዋ', 'ልትታደም', 'ሁለት', 'ልጆቿን', 'ወደ', 'ደቡብ', 'አፍ
ሪካ', 'ቢግ', 'ብራዘርስ', 'አፍሪካ', 'አብሮ', 'የመኖር', 'ውድድር', 'በደቡብ', 'አፍሪካ', 'ከተለያዩ', 'የአፍሪ
ካ', 'አገራት', 'የተውጣጡ', 'ያህል', 'ተሳታፊዎች', 'የተካፈሉበት', 'ይህ', 'ግለሰቦች', 'በፈታኝ', 'ሁኔታ',
'ውስጥ', 'በማለፍ', 'ብቃታቸውን', 'የሚያስመሰክሩበት', 'መሆኑን', 'የሚገጥሟቸውን', 'የተለያዩ', 'ፈተናዎች',
'በትእግስትና', 'በጥበብ', 'ከሌሎች', 'ጋር', 'ተስማምቶ', 'ችግሮችን', 'በብልጠት', 'መፍታት', 'ወዘተ', 'በየጊ
ዜው', 'ከሚደረገው', 'ቅነሳ', 'ተርፈው', 'ቀናት', 'ያህል', 'በውድድሩ', 'መቆየት', 'የቻሉ', 'ሁለት', 'ተወዳ
ዳሪዎች', 'እያንዳንዳቸው', 'ሺህ', 'ዶላር', 'እንደሚሸለሙም', 'ሲናገር', 'በዘንድሮው', 'ውድድር', 'አገራችን',
'ዳኒ', 'እና', 'ሃኒ', 'የተባሉ', 'ሁለት', 'ወጣቶችን', 'ብታሰልፍም', 'ዳኒ', 'ቀደም', 'ብሎ', 'የቅነሳው',
'ሰለባ', 'ሲሆን', 'ሃኒም', 'በቅርቡ', 'ከውድድር', 'ውጭ', 'የአገሪቱ', 'ብቸኛ', 'ተስፋ', 'ወደ', 'አሸናፊነ
ት', 'ለማሸጋገር', 'የህዝብ', 'የድጋፍ', 'ድም', 'ወሳኝ', 'መሆኑን', 'የተገነዘበው', 'ወዳጄ', 'ነው', 'እንግዲ
ህ', 'የሚል', 'አገራዊ', 'ጥሪ', 'ያስተላለፈልኝ', 'ያኔ', 'ሃኒ', 'ከውድድሩ', 'ከመሰናበቷ', 'የአገሩን', 'ስም',
'በአሸናፊነት', 'የማስጠራት', 'ከፍተኛ', 'አገሬ', 'እንዳትሸነፍ', 'የሚል', 'ከፍተኛ', 'ስጋት', 'እንዳደረበት', 'ጉ
ጉቱ', 'ሳይሆን', 'ስጋቱ', 'የወዳጄን', 'የዋህነት', 'ኢትዮጵያም', 'ይሸነፉ', 'ይሆን', 'በሚል', 'እንዲህ', 'ከን
ቱ', 'ስጋት']

Token size befor cleaned:  80769388

The first 50 words from the original tokens
 ['ምን', 'መሰላችሁ', 'አንባቢያን', 'ኢትዮጵያ', 'በተደጋጋሚ', 'ጥሪው', 'ደርሷት', 'ልትታደመው', 'ያልቻሉች
ው', 'የአለም', 'የእግር', 'ኳስ', 'ዋ', 'ለ19ኛ', 'ጊዜ', 'በደቡብ', 'አፍሪካ', 'ሲጠጣ፣', 'በሩቅ', 'እያየ
ች', 'አንጀቷ', 'ባረረ', 'ልክ', 'በአመቱ', 'በለስ', 'ቀናትና', 'ሌላ', 'ዋ', 'ልትታደም', 'ሁለት', 'ልጆቿ
ን', 'ወደ', 'ደቡብ', 'አፍሪካ', 'ላከች።6ኛው', 'ቢግ', 'ብራዘርስ', 'አፍሪካ', 'አብሮ', 'የመኖር', 'ውድድ
ር', 'በደቡብ', 'አፍሪካ', 'ተካሂዷል።', 'ከተለያዩ', '14', 'የአፍሪካ', 'አገራት', 'የተውጣጡ', '26', 'ያህ
ል', 'ተሳታፊዎች', 'የተካፈሉበት', 'ይህ', 'ውድድር፣', 'ግለሰቦች', 'በፈታኝ', 'ሁኔታ', 'ውስጥ', 'በማለፍ',
'ብቃታቸውን', 'የሚያስመሰክሩበት', 'መሆኑን', 'ሰምተናል።', 'የሚገጥሟቸውን', 'የተለያዩ', 'ፈተናዎች', 'በትእግስ
ትና', 'በጥበብ', 'ማለፍ፣', 'ከሌሎች', 'ጋር', 'ተስማምቶ', 'መዝለቅ፣', 'ችግሮችን', 'በብልጠት', 'መፍታት',
'ወዘተ', 'በየጊዜው', 'ከሚደረገው', 'ቅነሳ', 'ተርፈው', 'ለ91', 'ቀናት', 'ያህል', 'በውድድሩ', 'መቆየት',
'የቻሉ', 'ሁለት', 'ተወዳዳሪዎች', 'እያንዳንዳቸው', '200', 'ሺህ', 'ዶላር', 'እንደሚሸለሙም', 'ሲናገር', 'ነበ
ር።', 'በዘንድሮው', 'ውድድር', 'አገራችን', 'ዳኒ', 'እና', 'ሃኒ', 'የተባሉ', 'ሁለት', 'ወጣቶችን', 'ብታሰልፍ
ም', 'ዳኒ', 'ቀደም', 'ብሎ', 'የቅነሳው', 'ሰለባ', 'ሲሆን', 'ሃኒም', 'በቅርቡ', 'ከውድድር', 'ውጭ', 'ሆና
ለች።ይህቺን', 'የአገሪቱ', 'ብቸኛ', 'ተስፋ', 'ወደ', 'አሸናፊነት', 'ለማሸጋገር', 'የህዝብ', 'የድጋፍ', 'ድም',
'ወሳኝ', 'መሆኑን', 'የተገነዘበው', 'ወዳጄ', 'ነው', 'እንግዲህ', '835', 'የሚል', 'አገራዊ', 'ጥሪ', 'ያስተላለ
ፈልኝ', 'ያኔ', 'ሃኒ', 'ከውድድሩ', 'ከመሰናበቷ', 'በፊት።ወዳጄ', 'የአገሩን', 'ስም', 'በአሸናፊነት', 'የማስጠራ
ት', 'ከፍተኛ', 'ጉጉት፣', 'አገሬ']

Length difference/Number of removed tokens:  8956123
```

Save the cleaned tokens for later use

In [15]:
```
# Save the tokens for later use
with open('/kaggle/working/tokens_cleaned.pkl', 'wb') as file:
    pickle.dump(cleaned_tokens, file)
```

In [16]:
```
# # Load saved clean_Tokens, we have to load the token from different directory

with open('/kaggle/input/saved-tokens/tokens_cleaned(1).pkl', 'rb') as file:
    tokens = pickle.load(file)

# with open('/kaggle/working/tokens_cleaned.pkl', 'rb') as file:
#     tokens = pickle.load(file)
```

N-gram language mode

1.1 Create n-grams for n=1, 2, 3, 4.

- Top

In [19]:
```
# Extract n-grams n-gram generators and change to list
start_time = time.time()
unigrams = ngrams(tokens, 1)
bigrams = ngrams(tokens, 2)
trigrams = ngrams(tokens, 3)
fourgrams = ngrams(tokens, 4)
end_time = time.time()
print(f"\n\nTime taken for  n-gram generatoin :  {(end_time - start_time)/60}  minu


# we can change it to list but it is not memory efficient
unigramslist = list(ngrams(tokens, 1))
bigramslist = list(ngrams(tokens, 2))
trigramslist = list(ngrams(tokens, 3))
fourgramslist = list(ngrams(tokens, 4))
```

```
Time taken for  n-gram generatoin :  2.5510787963867186e-06  minutes
```

In [20]:
```
print(fourgramslist[:10])
```

```
[('ምን', 'መሰላችሁ', 'አንባቢያን', 'ኢትዮጵያ'), ('መሰላችሁ', 'አንባቢያን', 'ኢትዮጵያ', 'በተደጋጋሚ'), ('አንባቢያን', 'ኢትዮጵያ', 'በተደጋጋሚ', 'ጥሪው'), ('ኢትዮጵያ', 'በተደጋጋሚ', 'ጥሪው', 'ደርሷት'), ('በተደጋጋሚ', 'ጥሪው', 'ደርሷት', 'ልትታደመው'), ('ጥሪው', 'ደርሷት', 'ልትታደመው', 'ያልቻለችው'), ('ደርሷት', 'ልትታደመው', 'ያልቻለችው', 'የለም'), ('ልትታደመው', 'ያልቻለችው', 'የለም', 'የእግር'), ('ያልቻለችው', 'የለም', 'የእግር', 'ኳስ'), ('የለም', 'የእግር', 'ኳስ', 'ዋ')]
```

Print samples first 50 n-grams for all 4 n-grams

In [21]:
```
print('\n\nFirst 50 Unigrams\n')
print(unigramslist[:50])
print('\n')
print('\nFirst 50 Bigrams\n')
print(bigramslist[:50])
print('\n')
print('\nFirst 50 Trigrams\n')
print(trigramslist[:50])
```

```
print('\n')
print('\nFirst 50 fourgrams\n')
print(fourgramslist[:55])
```

First 50 Unigrams

[('ምን',), ('መሰላችሁ',), ('አንባቢያን',), ('ኢትዮጵያ',), ('በተደጋጋሚ',), ('ጥሪው',), ('ደርሷት',), ('ልትታደመው',), ('ያልቻለችው',), ('የአለም',), ('የእግር',), ('ኳስ',), ('ዋ',), ('ጊዜ',), ('በደቡብ',), ('አፍሪካ',), ('በሩቅ',), ('እያየች',), ('አንጀቷ',), ('ባረረ',), ('ልክ',), ('በአመቱ',), ('በለስ',), ('ቀናትና',), ('ሌላ',), ('ዋ',), ('ልትታደም',), ('ሁለት',), ('ልጆቿን',), ('ወደ',), ('ደቡብ',), ('አፍሪካ',), ('ቢግ',), ('ብራዘርስ',), ('አፍሪካ',), ('አብሮ',), ('የመኖር',), ('ውድድር',), ('በደቡብ',), ('አፍሪካ',), ('ከተለያዩ',), ('የአፍሪካ',), ('አገራት',), ('የተውጣጡ',), ('ያህል',), ('ተሳታፊዎች',), ('የተካፈሉበት',), ('ይህ',), ('ግለሰቦች',), ('በፈታኝ',)]

First 50 Bigrams

[('ምን', 'መሰላችሁ'), ('መሰላችሁ', 'አንባቢያን'), ('አንባቢያን', 'ኢትዮጵያ'), ('ኢትዮጵያ', 'በተደጋጋሚ'), ('በተደጋጋሚ', 'ጥሪው'), ('ጥሪው', 'ደርሷት'), ('ደርሷት', 'ልትታደመው'), ('ልትታደመው', 'ያልቻለችው'), ('ያልቻለችው', 'የአለም'), ('የአለም', 'የእግር'), ('የእግር', 'ኳስ'), ('ኳስ', 'ዋ'), ('ዋ', 'ጊዜ'), ('ጊዜ', 'በደቡብ'), ('በደቡብ', 'አፍሪካ'), ('አፍሪካ', 'በሩቅ'), ('በሩቅ', 'እያየች'), ('እያየች', 'አንጀቷ'), ('አንጀቷ', 'ባረረ'), ('ባረረ', 'ልክ'), ('ልክ', 'በአመቱ'), ('በአመቱ', 'በለስ'), ('በለስ', 'ቀናትና'), ('ቀናትና', 'ሌላ'), ('ሌላ', 'ዋ'), ('ዋ', 'ልትታደም'), ('ልትታደም', 'ሁለት'), ('ሁለት', 'ልጆቿን'), ('ልጆቿን', 'ወደ'), ('ወደ', 'ደቡብ'), ('ደቡብ', 'አፍሪካ'), ('አፍሪካ', 'ቢግ'), ('ቢግ', 'ብራዘርስ'), ('ብራዘርስ', 'አፍሪካ'), ('አፍሪካ', 'አብሮ'), ('አብሮ', 'የመኖር'), ('የመኖር', 'ውድድር'), ('ውድድር', 'በደቡብ'), ('በደቡብ', 'አፍሪካ'), ('አፍሪካ', 'ከተለያዩ'), ('ከተለያዩ', 'የአፍሪካ'), ('የአፍሪካ', 'አገራት'), ('አገራት', 'የተውጣጡ'), ('የተውጣጡ', 'ያህል'), ('ያህል', 'ተሳታፊዎች'), ('ተሳታፊዎች', 'የተካፈሉበት'), ('የተካፈሉበት', 'ይህ'), ('ይህ', 'ግለሰቦች'), ('ግለሰቦች', 'በፈታኝ'), ('በፈታኝ', 'ሁኔታ')]

First 50 Trigrams

[('ምን', 'መሰላችሁ', 'አንባቢያን'), ('መሰላችሁ', 'አንባቢያን', 'ኢትዮጵያ'), ('አንባቢያን', 'ኢትዮጵያ', 'በተደጋጋሚ'), ('ኢትዮጵያ', 'በተደጋጋሚ', 'ጥሪው'), ('በተደጋጋሚ', 'ጥሪው', 'ደርሷት'), ('ጥሪው', 'ደርሷት', 'ልትታደመው'), ('ደርሷት', 'ልትታደመው', 'ያልቻለችው'), ('ልትታደመው', 'ያልቻለችው', 'የአለም'), ('ያልቻለችው', 'የአለም', 'የእግር'), ('የአለም', 'የእግር', 'ኳስ'), ('የእግር', 'ኳስ', 'ዋ'), ('ኳስ', 'ዋ', 'ጊዜ'), ('ዋ', 'ጊዜ', 'በደቡብ'), ('ጊዜ', 'በደቡብ', 'አፍሪካ'), ('በደቡብ', 'አፍሪካ', 'በሩቅ'), ('አፍሪካ', 'በሩቅ', 'እያየች'), ('በሩቅ', 'እያየች', 'አንጀቷ'), ('እያየች', 'አንጀቷ', 'ባረረ'), ('አንጀቷ', 'ባረረ', 'ልክ'), ('ባረረ', 'ልክ', 'በአመቱ'), ('ልክ', 'በአመቱ', 'በለስ'), ('በአመቱ', 'በለስ', 'ቀናትና'), ('በለስ', 'ቀናትና', 'ሌላ'), ('ቀናትና', 'ሌላ', 'ዋ'), ('ሌላ', 'ዋ', 'ልትታደም'), ('ዋ', 'ልትታደም', 'ሁለት'), ('ልትታደም', 'ሁለት', 'ልጆቿን'), ('ሁለት', 'ልጆቿን', 'ወደ'), ('ልጆቿን', 'ወደ', 'ደቡብ'), ('ወደ', 'ደቡብ', 'አፍሪካ'), ('ደቡብ', 'አፍሪካ', 'ቢግ'), ('አፍሪካ', 'ቢግ', 'ብራዘርስ'), ('ቢግ', 'ብራዘርስ', 'አፍሪካ'), ('ብራዘርስ', 'አፍሪካ', 'አብሮ'), ('አፍሪካ', 'አብሮ', 'የመኖር'), ('አብሮ', 'የመኖር', 'ውድድር'), ('የመኖር', 'ውድድር', 'በደቡብ'), ('ውድድር', 'በደቡብ', 'አፍሪካ'), ('በደቡብ', 'አፍሪካ', 'ከተለያዩ'), ('አፍሪካ', 'ከተለያዩ', 'የአፍሪካ'), ('ከተለያዩ', 'የአፍሪካ', 'አገራት'), ('የአፍሪካ', 'አገራት', 'የተውጣጡ'), ('አገራት', 'የተውጣጡ', 'ያህል'), ('የተውጣጡ', 'ያህል', 'ተሳታፊዎች'), ('ያህል', 'ተሳታፊዎች', 'የተካፈሉበት'), ('ተሳታፊዎች', 'የተካፈሉበት', 'ይህ'), ('የተካፈሉበት', 'ይህ', 'ግለሰቦች'), ('ይህ', 'ግለሰቦች', 'በፈታኝ'), ('ግለሰቦች', 'በፈታኝ', 'ሁኔታ'), ('በፈታኝ', 'ሁኔታ', 'ውስጥ')]

First 50 fourgrams

[('ምን', 'መሰላችሁ', 'አንባቢያን', 'ኢትዮጵያ'), ('መሰላችሁ', 'አንባቢያን', 'ኢትዮጵያ', 'በተደጋጋሚ'), ('አንባቢያን', 'ኢትዮጵያ', 'በተደጋጋሚ', 'ጥሪው'), ('ኢትዮጵያ', 'በተደጋጋሚ', 'ጥሪው', 'ደርሷት'), ('በተደጋጋሚ', 'ጥሪው', 'ደርሷት', 'ልትታደመው'), ('ጥሪው', 'ደርሷት', 'ልትታደመው', 'ያልቻለችው'), ('ደርሷት', 'ልትታደመው', 'ያልቻለችው', 'የአለም'), ('ልትታደመው', 'ያልቻለችው', 'የአለም', 'የእግር'), ('ያልቻለችው', 'የአለም', 'የእግር', 'ኳስ'), ('የአለም', 'የእግር', 'ኳስ', 'ዋ'), ('የእግር', 'ኳስ', 'ዋ', 'ጊዜ'), ('ኳስ', 'ዋ', 'ጊዜ', 'በደቡብ'), ('ዋ', 'ጊዜ', 'በደቡብ', 'አፍሪካ'), ('ጊዜ', 'በደቡብ', 'አፍሪካ', 'በሩቅ'), ('በደቡብ', 'አፍሪካ', 'በሩቅ', 'እያየች'), ('አፍሪካ', 'በሩቅ', 'እያየች', 'አንጀቷ'), ('በሩቅ', 'እያየች', 'አንጀቷ', 'ባረረ'), ('እያየች', 'አንጀቷ', 'ባረረ', 'ልክ'), ('አንጀቷ', 'ባረረ', 'ልክ', 'በአመቱ'), ('ባረረ', 'ልክ', 'በአመቱ', 'በለስ'), ('ልክ', 'በአመቱ', 'በለስ', 'ቀናትና'), ('በአመቱ', 'በለስ', 'ቀናትና', 'ሌላ'), ('በለስ', 'ቀናትና', 'ሌላ', 'ዋ'), ('ቀናትና', 'ሌላ', 'ዋ', 'ልትታደም'), ('ሌላ', 'ዋ', 'ልትታደም', 'ሁለት'), ('ዋ', 'ልትታደም', 'ሁለት', 'ልጆቿን'), ('ልትታደም', 'ሁለት', 'ልጆቿን', 'ወ

ደ'), ('ሁለት', 'ልጆቿን', 'ወደ', 'ደቡብ'), ('ልጆቿን', 'ወደ', 'ደቡብ', 'አፍሪካ'), ('ወደ', 'ደቡብ', 'አፍሪካ', 'ቢግ'), ('ደቡብ', 'አፍሪካ', 'ቢግ', 'ብራዘርስ'), ('አፍሪካ', 'ቢግ', 'ብራዘርስ', 'አፍሪካ'), ('ቢግ', 'ብራዘርስ', 'አፍሪካ', 'አብሮ'), ('ብራዘርስ', 'አፍሪካ', 'አብሮ', 'የመኖር'), ('አፍሪካ', 'አብሮ', 'የመኖር', 'ውድድር'), ('አብሮ', 'የመኖር', 'ውድድር', 'በደቡብ'), ('የመኖር', 'ውድድር', 'በደቡብ', 'አፍሪካ'), ('ውድድር', 'በደቡብ', 'አፍሪካ', 'ከተለያዩ'), ('በደቡብ', 'አፍሪካ', 'ከተለያዩ', 'የአፍሪካ'), ('አፍሪካ', 'ከተለያዩ', 'የአፍሪካ', 'አገራት'), ('ከተለያዩ', 'የአፍሪካ', 'አገራት', 'የተውጣጡ'), ('የአፍሪካ', 'አገራት', 'የተውጣጡ', 'ያህል'), ('አገራት', 'የተውጣጡ', 'ያህል', 'ተሳታፊዎች'), ('የተውጣጡ', 'ያህል', 'ተሳታፊዎች', 'የተካፈሉበት'), ('ያህል', 'ተሳታፊዎች', 'የተካፈሉበት', 'ይህ'), ('ተሳታፊዎች', 'የተካፈሉበት', 'ይህ', 'ግለሰቦች'), ('የተካፈሉበት', 'ይህ', 'ግለሰቦች', 'በፈታኝ'), ('ይህ', 'ግለሰቦች', 'በፈታኝ', 'ሁኔታ'), ('ግለሰቦች', 'በፈታኝ', 'ሁኔታ', 'ውስጥ'), ('በፈታኝ', 'ሁኔታ', 'ውስጥ', 'በማለፍ'), ('ሁኔታ', 'ውስጥ', 'በማለፍ', 'ብቃታቸውን'), ('ውስጥ', 'በማለፍ', 'ብቃታቸውን', 'የሚያስመሰክሩበት'), ('በማለፍ', 'ብቃታቸውን', 'የሚያስመሰክሩበት', 'መሆኑን'), ('ብቃታቸውን', 'የሚያስመሰክሩበት', 'መሆኑን', 'የሚገጥሟቸውን'), ('የሚያስመሰክሩበት', 'መሆኑን', 'የሚገጥሟቸውን', 'የተለያዩ')]

In [22]:
```
# sample prints
# print samples for all 4 n-grams
print('\nUnigrams')
print(next(unigrams))
print(next(unigrams))
print(next(unigrams))
print(next(unigrams))

print('\nBigrams')
print(next(bigrams))
print(next(bigrams))
print(next(bigrams))
print(next(bigrams))

print('\nTrigrams')
print(next(trigrams))
print(next(trigrams))
print(next(trigrams))
print(next(trigrams))
print('\nFourgrams')

print(next(fourgrams))
print(next(fourgrams))
print(next(fourgrams))
print(next(fourgrams))
```

```
Unigrams
('ምን',)
('መሰላችሁ',)
('አንባቢያን',)
('ኢትዮጵያ',)

Bigrams
('ምን', 'መሰላችሁ')
('መሰላችሁ', 'አንባቢያን')
('አንባቢያን', 'ኢትዮጵያ')
('ኢትዮጵያ', 'በተደጋጋሚ')

Trigrams
('ምን', 'መሰላችሁ', 'አንባቢያን')
('መሰላችሁ', 'አንባቢያን', 'ኢትዮጵያ')
('አንባቢያን', 'ኢትዮጵያ', 'በተደጋጋሚ')
('ኢትዮጵያ', 'በተደጋጋሚ', 'ጥሪው')

Fourgrams
('ምን', 'መሰላችሁ', 'አንባቢያን', 'ኢትዮጵያ')
('መሰላችሁ', 'አንባቢያን', 'ኢትዮጵያ', 'በተደጋጋሚ')
('አንባቢያን', 'ኢትዮጵያ', 'በተደጋጋሚ', 'ጥሪው')
('ኢትዮጵያ', 'በተደጋጋሚ', 'ጥሪው', 'ደርሷት')
```

1.2. Calculate probabilities of n-grams and find the top 10 most likely n-grams for all n.

- Top

1.2.1. probabilities of 2-grams and top 10 most likely 2-grams

Calculate probabilities

In [23]:
```
# Get frequency of bigrams
bigram_freq = Counter(bigrams)
trigram_freq = Counter(trigrams)
fourgram_freq = Counter(fourgrams)

# # Calculate probabilities
total_bigrams = sum(bigram_freq.values())
bigram_probabilities = {gram: freq/total_bigrams for gram, freq in bigram_freq.item
```

Get top 10 bigrams

In [24]:
```
top_10_bigrams = sorted(bigram_probabilities.items(), key=lambda item: item[1], rev
print("{:<40} {:<20}".format("\nTop 10 fourgrams ", " Probability"))

# Print each element in the desired format
for ngram, probability in top_10_bigrams:
    ngram_str = ' '.join(ngram)
    print("{:<40} {:<20}".format(ngram_str, probability))
```

Top 10 fourgrams	Probability
ዓ ም	0.0018562964337997358
ነገር ግን	0.0007995556147818973
ቀን ዓ	0.0007798519096186599
ብቻ ሳይሆን	0.00041674077161042754
አዲስ አበባ	0.00038229632461454256
ኤ አ	0.00037214817571467967
ምክር ቤት	0.00037207410163511863
በአዲስ አበባ	0.00036607410119067415
እ ኤ	0.00035088891488066037
ምን ያህል	0.0003265926167846383

1.2.2. probabilities of trigrams and top 10 most likely trigrams

Calculate probabilities

In [25]:
```
total_trigrams = sum(trigram_freq.values())
trigram_probabilities = {gram: freq/total_trigrams for gram, freq in trigram_freq.i
```

Get top 10 trigrams

In [26]:
```
top_10_trigrams = sorted(trigram_probabilities.items(), key=lambda item: item[1], r
print("{:<40} {:<20}".format("\nTop 10 fourgrams ", " Probability"))
# Print each element in the desired format
for ngram, probability in top_10_trigrams:
    ngram_str = ' '.join(ngram)
    print("{:<40} {:<20}".format(ngram_str, probability))
```

```
Top 10 fourgrams                          Probability
ቀን ዓ ም                                   0.000776148263133076
እ ኤ አ                                    0.00034837042198080325
ዓ ም ጀምሮ                                 0.00011970372143758836
ተወካዮች ምክር ቤት                          0.00010162964468587329
አበባ ከተማ አስተዳደር                       8.792593895199096e-05
የአዲስ አበባ ከተማ                          8.214816031824597e-05
በዓለም አቀፍ ደረጃ                          6.70370469684514e-05
ከጊዜ ወደ ጊዜ                              6.548149118244314e-05
ጥር ቀን ዓ                                 6.540741709739512e-05
የካቲት ቀን ዓ                              6.496297258710705e-05
```

1.2.3. probabilities of 4-grams and top 10 most likely 4-grams

Calculate probabilities

In [27]:
```
total_fourgrams = sum(fourgram_freq.values())
fourgrams_probabilities = {gram: freq/total_fourgrams for gram, freq in fourgram_fr
```

Get top 10 fourgrams

In [28]:
```
top_10_fourgrams = sorted(fourgrams_probabilities.items(), key=lambda item: item[1]
print("{:<40} {:<20}".format("\nTop 10 fourgrams ", "  Probability"))

# Print each element in the desired format
for ngram, probability in top_10_fourgrams:
    ngram_str = ' '.join(ngram)
    print("{:<40} {:<20}".format(ngram_str, probability))
```

```
Top 10 fourgrams                          Probability
ጥር ቀን ዓ ም                               6.503705148971514e-05
የካቲት ቀን ዓ ም                           6.48148292181102e-05
መጋቢት ቀን ዓ ም                           6.25926065020607e-05
ጥቅምት ቀን ዓ ም                           5.311112291358287e-05
ግንቦት ቀን ዓ ም                           5.222223382716307e-05
አመለካከት ብቻ የሚያንፀባርቅ መሆኑን           5.214815973662809e-05
የጸሐፊውን አመለካከት ብቻ የሚያንፀባርቅ          5.029630747325351e-05
ጽሑፉ የጸሐፊውን አመለካከት ብቻ                5.0148159292183544e-05
ቀን ዓ ም ጀምሮ                             4.977778883950863e-05
ሰኔ ቀን ዓ ም                              4.851852930041392e-05
```

1.3. What is the probability of the sentence. "ኢትዮጵያ ታሪካዊ ሀገር ናት ".

- Top

In [29]:
```
def generate_ngrams(tokens, n):
    return list(ngrams(tokens, n))


#n-gram_probabilities is a dictionary containing n-gram probabilities
def probability_of_sentence(sentence,n):
    sentence_tokens = sentence.split()
    ngrams_sentence = generate_ngrams(sentence_tokens, n)
    sentence_probability = 1.0

    if n==2:
        for bg in ngrams_sentence:
```

```
            if bg in bigram_probabilities:
                sentence_probability *= bigram_probabilities[bg]
            else:
                sentence_probability *= 0.0  # Assuming we assign zero probability
        return sentence_probability
    elif n==3:
        for bg in ngrams_sentence:
            if bg in trigram_probabilities:
                sentence_probability *= trigram_probabilities[bg]
            else:
                sentence_probability *= 0.0  # Assuming we assign zero probability
        return sentence_probability
    elif n==4:
        for bg in ngrams_sentence:
            if bg in fourgrams_probabilities:
                sentence_probability *= fourgrams_probabilities[bg]
            else:
                sentence_probability *= 0.0  # Assuming we assign zero probability
        return sentence_probability
    else:
        return 'Please input n 2, 3, or 4'

sentence1 = "ኢትዮጵያ ታሪካዊ ሀገር ናት"



print(f"\n\n\nThe probability of the sentence(ኢትዮጵያ ታሪካዊ ሀገር ናት) using bigram is:

print(f"\nThe probability of the sentence(ኢትዮጵያ ታሪካዊ ሀገር ናት) using trigram is:

print(f"\nThe probability of the sentence(ኢትዮጵያ ታሪካዊ ሀገር ናት) using fourgram is:
```

```
The probability of the sentence(ኢትዮጵያ ታሪካዊ ሀገር ናት) using bigram is:    4.389575735
4063326e-20

The probability of the sentence(ኢትዮጵያ ታሪካዊ ሀገር ናት) using trigram is:    5.48697007
5700204e-15

The probability of the sentence(ኢትዮጵያ ታሪካዊ ሀገር ናት) using fourgram is:    7.4074090
53498309e-08
```

`Bigram Model (n=2)` : The probability is incredibly low at approximately 4.39e-20. This suggests that, according to the bigram model, the chance of this exact sequence of words appearing together is extremely rare within the context of the text corpus used for the model.

`Trigram Model (n=3)` : The probability increases significantly to around 5.49e-15 when using a trigram model. This implies that considering a sequence of three words at a time provides a higher likelihood for this sentence to be observed, reflecting a more common occurrence in the source material.

`Fourgram Model (n=4)` : The probability surges to approximately 7.41e-08 when evaluated with a fourgram model. This indicates that the sequence of four words is much more common, and the sentence is much more likely to occur in the dataset that was used to train the model.

The increasing probabilities of the Amharic sentence " `ኢትዮጵያ ታሪካዊ ሀገር ናት` " with larger n-grams from bigrams to fourgrams demonstrate the importance of context in language

modeling. The low probability with the bigram model indicates a rare occurrence of word pairs, while the higher probabilities with trigram and fourgram models suggest that longer word sequences from the sentence are more common in the dataset. This trend highlights the need for extensive and context-rich datasets to train accurate language models, especially for languages with complex structures like Amharic.

a) using Bigrams

In [30]:
```
sentence1 = "ኢትዮጵያ ታሪካዊ ሀገር ናት"
sentence2 = "በፈታኝ ሁኔታ ውስጥ ማለፍ"
sentence3 = 'ተወካዮች ምክር ቤት'
sentence4 = 'በዓለም አቀፍ ደረጃ'
sentence5 = 'በአዲስ አበባ'


print(f"\n\n\nThe probability of the sentence(ኢትዮጵያ ታሪካዊ ሀገር ናት) is: \
{probability_of_sentence(sentence1, 2)}")
print(f"\nThe probability of the sentence(በፈታኝ ሁኔታ ውስጥ ማለፍ) is:  \
{probability_of_sentence(sentence2, 2)}")
print(f"\nThe probability of the sentence(ተወካዮች ምክር ቤት) is:   \
{probability_of_sentence(sentence3, 2)}")
print(f"\nThe probability of the sentence(በዓለም አቀፍ ደረጃ) is:   \
{probability_of_sentence(sentence4, 2)}")
print(f"\nThe probability of the sentence(በአዲስ አበባ) is:   \
{probability_of_sentence(sentence5, 2)}")
```

```
The probability of the sentence(ኢትዮጵያ ታሪካዊ ሀገር ናት) is: 4.3895757354063326e-20

The probability of the sentence(በፈታኝ ሁኔታ ውስጥ ማለፍ) is:  1.0239904729428439e-17

The probability of the sentence(ተወካዮች ምክር ቤት) is:   4.169986351657763e-08

The probability of the sentence(በዓለም አቀፍ ደረጃ) is:   1.7296891451391234e-08

The probability of the sentence(በአዲስ አበባ) is:   0.00036607410119067415
```

b) Using trigrams

In [31]:
```
sentence1 = "ኢትዮጵያ ታሪካዊ ሀገር ናት"
sentence2 = "በፈታኝ ሁኔታ ውስጥ ማለፍ"
sentence3 = 'ተወካዮች ምክር ቤት'
sentence4 = 'በዓለም አቀፍ ደረጃ'


print(f"\n\nThe probability of the sentence(ኢትዮጵያ ታሪካዊ ሀገር ናት) is: \
{probability_of_sentence(sentence1, 3)}")
print(f"\nThe probability of the sentence(በፈታኝ ሁኔታ ውስጥ ማለፍ) is:  \
{probability_of_sentence(sentence2, 3)}")

print(f"\nThe probability of the sentence(ተወካዮች ምክር ቤት) is:   \
{probability_of_sentence(sentence3, 3)}")
print(f"\nThe probability of the sentence(በዓለም አቀፍ ደረጃ) is:   \
{probability_of_sentence(sentence4, 3)}")
```

```
The probability of the sentence(ኢትዮጵያ ታሪካዊ ሀገር ናት) is: 5.486970075700204e-15

The probability of the sentence(በፈታኝ ሁኔታ ውስጥ ማለፍ) is:  0.0

The probability of the sentence(ተወካዮች ምክር ቤት) is:   0.00010162964468587329

The probability of the sentence(በዓለም አቀፍ ደረጃ) is:   6.70370469684514e-05
```

c) Using fourgrams

In [32]:
```
sentence1 = "ኢትዮጵያ ታሪካዊ ሀገር ናት"
sentence2 = "ግለሰቦች በፈታኝ ሁኔታ ውስጥ በማለፍ ብቃታቸውን የሚያስመሰክሩበት መሆኑን"
sentence3 = "በፈታኝ ሁኔታ ውስጥ በማለፍ"


print(f"\n\nThe probability of the sentence(ኢትዮጵያ ታሪካዊ ሀገር ናት) is:   \
{probability_of_sentence(sentence1, 4)}")
print(f"\nThe probability of the sentence(ግለሰቦች በፈታኝ ሁኔታ ውስጥ በማለፍ ብቃታቸውን የሚያስመሰ
{probability_of_sentence(sentence2, 4)}")

print(f"\nThe probability of the sentence(በፈታኝ ሁኔታ ውስጥ በማለፍ) is:   \
{probability_of_sentence(sentence3, 4)}")
```

```
The probability of the sentence(ኢትዮጵያ ታሪካዊ ሀገር ናት) is:   7.407409053498309e-08

The probability of the sentence(ግለሰቦች በፈታኝ ሁኔታ ውስጥ በማለፍ ብቃታቸውን የሚያስመሰክሩበት መሆኑ
ን) is:  2.2301374979696553e-36

The probability of the sentence(በፈታኝ ሁኔታ ውስጥ በማለፍ) is:   7.407409053498309e-08
```

1.4. Generate random sentences using n-grams; explain what happens as n-increases based on your output

- Top

Define random sentence generator function

In [33]:
```
def sentence_generator(gram_freq,n, token, num_words=8):
    if n==2:
        bigram_sentence = [token]
        for _ in range(num_words - 1):
            next_tokens = [pair[1] for pair in gram_freq if pair[0] == token]
            if not next_tokens:
                break
            token = random.choice(next_tokens)
            bigram_sentence.append(token)
        return ' '.join(bigram_sentence)
    elif n==3:
        trigram_sentence = [token]
        for _ in range(num_words - 1):
            next_tokens = [pair[1] for pair in gram_freq if pair[0] == token]
            if not next_tokens:
                break
            token = random.choice(next_tokens)
            trigram_sentence.append(token)
        return ' '.join(trigram_sentence)
    elif n==4:
```

```
        fourgram_sentence = [token]
        for _ in range(num_words - 1):
            next_tokens = [pair[1] for pair in gram_freq if pair[0] == token]
            if not next_tokens:
                break
            token = random.choice(next_tokens)
            fourgram_sentence.append(token)
        return ' '.join(fourgram_sentence)
    else:
        return 'Please input one of the following n values n=2,n=3, n=4'
```

a) Using bigrams

In [34]:
```
import random
# def generate_sentence(bigram_freq, token, num_words=8):
#     sentence = [token]
#     for _ in range(num_words - 1):
#         next_tokens = [pair[1] for pair in bigram_freq if pair[0] == token]
#         if not next_tokens:
#             break
#         token = random.choice(next_tokens)
#         sentence.append(token)
#     return ' '.join(sentence)

# Generate a sentence starting with a given word
starting_word1 = 'ኢትዮጵያ'
starting_word2 = 'በተደጋጋሚ'
starting_word3 = 'ከተለያዩ'
starting_word4 = 'የአለም'
starting_word5 = 'በደቡብ'
starting_word6 = 'የአፍሪካ'
starting_word7 = 'በሩቅ'



print('ኢትዮጵያ as a starting word =  ',sentence_generator(bigram_freq,2, starting_wo
print('በተደጋጋሚ as a starting word =  ',sentence_generator(bigram_freq,2,starting_wo
print('ከተለያዩ as a starting word = ',sentence_generator(bigram_freq,2,starting_word3
print('የአለም as a starting word = ',sentence_generator(bigram_freq,2,starting_word4)
print('በደቡብ as a starting word =  ',sentence_generator(bigram_freq,2,starting_word
print('የአፍሪካ as a starting word = ',sentence_generator(bigram_freq,2,starting_word6
print('በሩቅ as a starting word = ',sentence_generator(bigram_freq,2,starting_word7))
```

```
ኢትዮጵያ as a starting word =  ኢትዮጵያ ውሥጥ ጣሊያኖች የቅኝ ድንበር መመለስ የፈለግን እኔ
በተደጋጋሚ as a starting word =  በተደጋጋሚ የሚቋቋሙበትና መደበኛ ጉባኤው በቀረበው ምርት ኦፐሬሽን ጁባ
ከተለያዩ as a starting word =  ከተለያዩ ምግቦች ውስጥ አድናቆት እነዛኞቹ ደግሞ ምናለበት ይህንን
የአለም as a starting word =  የአለም ቅርሶችን ግእዝ የኢትዮጵያውያን ባህልና ልማድ ከተጨባጩ የኢኮኖሚ
በደቡብ as a starting word =  በደቡብ አሜሪካዋ ቬኒዙዌላ ወይም የእጩ ፕሬዚዳንት አንዳንድ የብረት
የአፍሪካ as a starting word =  የአፍሪካ ቀዳሚ መመዘኛ ተደርጎ አሸናፊውን አለቀ ስለዘህ እንደተጠቃሚው
በሩቅ as a starting word =  በሩቅ ሰምቱን በልቦና ማላመጥ አዝማሪውንም ውጦ ቅምም ወጨፎ
```

b) Using trigrams

In [35]:
```
# Generate a sentence starting with a given word
starting_word1 = 'ኢትዮጵያ'
starting_word2 = 'በተደጋጋሚ'
starting_word3 = 'ከተለያዩ'
starting_word4 = 'የአለም'
starting_word5 = 'በደቡብ'
starting_word6 = 'የአፍሪካ'
```

```
starting_word7 = 'በሩቅ'



print('ኢትዮጵያ as a starting word =  ',sentence_generator(trigram_freq,3, starting_w
print('በተደጋጋሚ as a starting word =  ',sentence_generator(trigram_freq,3, starting_
print('ከተለያዩ as a starting word = ',sentence_generator(trigram_freq,3, starting_wor
print('የአለም as a starting word = ',sentence_generator(trigram_freq,3, starting_word
print('በደቡብ as a starting word =  ',sentence_generator(trigram_freq,3, starting_wo
print('የአፍሪካ as a starting word = ',sentence_generator(trigram_freq,3, starting_wor
print('በሩቅ as a starting word = ',sentence_generator(trigram_freq,3, starting_word7
```

```
ኢትዮጵያ as a starting word =  ኢትዮጵያ ከመምጣታችን ዘግየት ብሎ ታኪታ ዋጋ ደግሞ በአሣታሚዎችም
በተደጋጋሚ as a starting word =  በተደጋጋሚ ወረራ ሳቢያ የወሰዱትን በሚሊዮን የሚቆጠር መሆኑን ብናውቅም
ከተለያዩ as a starting word =  ከተለያዩ አካባቢዎች ሦስት ከንጋቱ ሰዓት ላይ ማሻሻያ የሚጨምር
የአለም as a starting word =  የአለም ጦርነት ያህል ተበድላ እንደኖረች በመጽሐፉ ውስጥ መግባት
በደቡብ as a starting word =  በደቡብ ዐረብ ነን ከሚሉት ውስጥ ሆነን ቀረን ስልሳ
የአፍሪካ as a starting word =  የአፍሪካ አገሮች ያሏቸውን ጉዳዮች እውቅና መስጠት ሁሉም ባለሥልጣናት
በሩቅ as a starting word =  በሩቅ በሆያ ሆዬ አዳም በመሆኑ አንድን የደርግ ወታደሮችና
```

c) Using fourgrams

In [36]:
```
starting_word1 = 'ኢትዮጵያ'
starting_word2 = 'በተደጋጋሚ'
starting_word3 = 'ከተለያዩ'
starting_word4 = 'የአለም'
starting_word5 = 'በደቡብ'
starting_word6 = 'የአፍሪካ'
starting_word7 = 'በሩቅ'



print('ኢትዮጵያ as a starting word =  ',sentence_generator(fourgram_freq,4, starting_
print('በተደጋጋሚ as a starting word =  ',sentence_generator(fourgram_freq,4, starting
print('ከተለያዩ as a starting word = ',sentence_generator(fourgram_freq,4, starting_wo
print('የአለም as a starting word = ',sentence_generator(fourgram_freq,4, starting_wor
print('በደቡብ as a starting word =  ',sentence_generator(fourgram_freq,4, starting_w
print('የአፍሪካ as a starting word = ',sentence_generator(fourgram_freq,4, starting_wo
print('በሩቅ as a starting word = ',sentence_generator(fourgram_freq,4, starting_word
```

```
ኢትዮጵያ as a starting word =  ኢትዮጵያ ቀሚስ በጭንቅላቱ የሚመላለሰው ምንድነው ያለው ዕርቅ እንዲኖር
በተደጋጋሚ as a starting word =  በተደጋጋሚ ተመሳሳይ አቋም ከያዙ ወዲህ በቲማቲም ቆርቆሮ በቆርቆሮ
ከተለያዩ as a starting word =  ከተለያዩ የፍትሕ ቀን እሱ ላይ የላቀ ድል በሁለተኛው
የአለም as a starting word =  የአለም ታላላቅ ኢትዮጵያውያን ወንድሞቻቸው ጎን ለጎን ዓለማዊ ዝማጅታቸው
በደቡብ as a starting word =  በደቡብ አረቢያ እየተባረሩ ይህ ዞን ቀበሌዎች ውስጥ ያሉ
የአፍሪካ as a starting word =  የአፍሪካ አየር መንገዱ ግራና ቀኝ ሲኦልና የአፍሪካ አገሮች
በሩቅ as a starting word =  በሩቅ ላለችው አባወራ ሦስት ዓመታት መልካም አስተዳደሮችን አንባገነናዊ
```

2. Evaluate these Language Models Using Intrinsic Evaluation Method.

Perplexity

In order to implement the perplexity formula, you'll need to know how to implement m-th order root of a variable.

$$PP(W) = \sqrt[M]{\prod_{i=1}^{m} \frac{1}{P(w_i|w_{i-1})}}$$

From calculus:

$$\sqrt[M]{\frac{1}{x}} = x^{-\frac{1}{M}}$$

Here is bellow a code that will help us with the formula.

- Top

In [37]:
```python
import math
from collections import Counter

def calculate_perplexity(ngrams, total_words):
    model_entropy = 0.0
    for ngram, count in ngrams.items():
        prob = count / total_words
        model_entropy += -math.log2(prob)

    model_entropy /= total_words
    return math.pow(2, model_entropy)

def evaluate_model(n, train_tokens, test_tokens):

    ngrams_train = Counter(tuple(train_tokens[i:i + n]) for i in range(len(train_to
    total_words_train = len(train_tokens)

    ngrams_test = Counter(tuple(test_tokens[i:i + n]) for i in range(len(test_toker
    total_words_test = len(test_tokens)

    return calculate_perplexity(ngrams_test, total_words_test)
# Load the tokens and calculate frequencies
with open('/kaggle/input/saved-tokens/tokens_cleaned(1).pkl', 'rb') as file:
    tokens = pickle.load(file)
# Split the data into training and test sets (80% training, 20% test)
train_size = int(0.8 * len(tokens))
train_tokens = tokens[:train_size]
test_tokens = tokens[train_size:]

# Evaluating n-gram models for n = 1, 2, 3, 4
perplexities = []
for n in range(1, 5):
    perplexity = evaluate_model(n, train_tokens, test_tokens)
    perplexities.append(perplexity)


# Storing the results in a list of tuples
results = []

for n, perplexity in enumerate(perplexities):
    results.append((n + 1, perplexity))

# Printing the results
for n, perplexity in results:
    print(f"Perplexity of {n}-gram model is  : {perplexity}")
```

```
Perplexity of 1-gram model is  : 3.4376775176273546
Perplexity of 2-gram model is  : 13141.287652919986
Perplexity of 3-gram model is  : 651342.603025604
Perplexity of 4-gram model is  : 1549724.6862948143
```

The perplexity values you obtained indicate how well our n-gram models generalize to unseen data. Perplexity is a measure of uncertainty or "surprise" associated with predicting the next word in a sequence. A lower perplexity indicates better performance.

Here's a brief interpretation of the results:

`Unigram (n=1):`

> Perplexity: 3.44 The unigram model has relatively low perplexity, suggesting that it provides a good estimate of word probabilities based on individual words. This is expected, as unigrams consider each word in isolation.

`Bigram (n=2):`

> Perplexity: 13141.29 The bigram model has a significantly higher perplexity compared to the unigram model. This indicates that predicting the next word based on the previous word alone might not be as accurate. It suggests that the model struggles with capturing dependencies between adjacent words.

`Trigram (n=3):`

> Perplexity: 651342.60 The trigram model has an even higher perplexity, indicating that considering the previous two words for prediction does not lead to effective generalization. The model might be overfitting to the training data and failing to capture broader linguistic patterns.

`Fourgram (n=4):`

> Perplexity: 1549724.69` The fourgram model exhibits the highest perplexity, suggesting that considering the previous three words for prediction may lead to overfitting or insufficient data for effective generalization. The model might struggle with unseen combinations of four words.

Description: The results highlight the trade-off between model complexity and generalization. While the unigram model performs well, higher-order n-gram models (bigram, trigram, fourgram) face challenges in capturing dependencies and suffer from increasing perplexity, indicating potential overfitting to the training data. This could be due to the limitations of n-gram models in handling the complexity of natural language, especially when dealing with longer dependencies between words. Consideration of more advanced language models or techniques may be warranted for improved performance on this task.

3. Evaluate these Language Models Using Extrinsic Evaluation Method

- Top

Create predictive model for extrinsic evaluation

In [38]:
```
# Function for predictive text entry (for extrinsic evaluation)
def predict_next_word(current_sentence, bigram_freq):
    current_tokens = current_sentence.split()
    last_token = current_tokens[-1]
    next_words = {pair[1]: freq for pair, freq in bigram_freq.items() if pair[0] ==
    next_word = max(next_words, key=next_words.get) if next_words else None
    return next_word
```

Extrinsic evaluation - predict next word

In [39]:
```
# Extrinsic evaluation - predict next word
current_sentence = "ታሪካዊ ሀገር"
next_word = predict_next_word(current_sentence, bigram_freq)
print('\n\nThe previous words are:  ',current_sentence)
print(f'\n and the suggested word is: {next_word}')
```

```
The previous words are:   ታሪካዊ ሀገር

 and the suggested word is: ውስጥ
```

In [40]:
```
current_sentence = "ታሪካዊ"
next_word = predict_next_word(current_sentence, bigram_freq)
print('\n\nThe previous words are:  ',current_sentence)
print(f'\n and the suggested word is: {next_word}')
```

```
The previous words are:   ታሪካዊ

 and the suggested word is: ዳራ
```

In [41]:
```
current_sentence = "በፈታኝ ሁኔታ"
next_word = predict_next_word(current_sentence, bigram_freq)
print('\n\nThe previous words are:  ',current_sentence)
print(f'\n and the suggested word is: {next_word}')
```

```
The previous words are:   በፈታኝ ሁኔታ

 and the suggested word is: ውስጥ
```

In [42]:
```
current_sentence = "ውስጥ በማለፍ"
next_word = predict_next_word(current_sentence, bigram_freq)
print('\n\nThe previous words are:  ',current_sentence)
print(f'\n and the suggested word is: {next_word}')
```

```
The previous words are:   ውስጥ በማለፍ

 and the suggested word is: ወደ
```

Conclusion

The model utilizes bigram frequencies to predict the next word in a sentence, and the given test cases show that it can suggest contextually reasonable continuations. For instance, from the historical context indicated by "ታሪካዊ ሀገር," the model aptly predicts "ውስጥ," reflecting a coherent phrase structure in the language being tested. The other test cases align similarly with the expected linguistic patterns. Thus, based on these examples, we can infer that the

model is fairly effective in its predictive capability, indicating a good understanding of word pair probabilities within the corpus from which the bigram frequencies were derived.