Assignment-1 Frequency Analysis and Bag of Words

Objective

- Learn the keywords of NLP
- Understand basic of NLP
- Explore the Bag of words for basic analysis of text

Your Details

```
import datetime

student_rollno = 24
student_name = 'Rachit Basnet'
assignment_tag = 'MDS555-2023-Assignment-1'


# from checker_utils import done
def done(task):
    _date = datetime.datetime.now()
    task = task + ": " + str(_date)
    print('='*len(task), '\n', task , '\n', '='*len(task), sep='')
    pass
```

Literature Review

- Put your review of the literature related to Frequency analysis and bag of words
- define terminologies used
- put details of the library used

Task 1: Dataset Preparation:

Prepare the Nepali news dataset *(hint: you can obtain text from news websites, at least 20 different news of 2/3 different categories)*. Host the dataset in the public git repository.

```
# Task 1:Dataset preparation:
import pandas as pd
!git clone https://github.com/rachitbasnet/Assignment.git
#Load dataset
df =pd.read_csv('/content/Assignment/news for nlp3.csv')
print(df)
```

```
fatal: destination path 'Assignment' already exists and is not an empty directory.
    Sn Catogory                                              News
0    1  Finance  मूल्यवृद्धिसँगै अब काठमाडौंमा पेट्रोल प्रतिलिट...
1    2  Finance  सरकारले पेट्रोलियम पदार्थको मूल्य फेरि बढाएको ...
2    3  Finance  हालसम्म ४ वाणिज्य बैंकहरुले गत आर्थिक वर्षको न...
3    4  Finance  चालु आर्थिक वर्षको पहिलो महिना साउनमा मुलुकबाट...
4    5  Opinion  नेपाली राजनीति र साहित्यमा सबैभन्दा धेरै एकैसा...
5    6  Opinion  पत्रकार रमेशकुमारले हिमालखबरमा अघिल्लो साता आक...
6    7  Opinion  बिरामीको उपचारमा लापरबाही गरेको आरोप लगाउँदै ब...
7    8  Opinion  नेपालको राजनीतिमा अचेल सबै ठूला दलका नेताहरू ए...
8    9  Finance  जिल्लाको उत्तरी भेगमा भएर बग्ने कालीगण्डकी नदी...
9   10  Finance   नेपाल धितोपत्र बोर्ड (सेबोन) ले ब्रोकर कमिसन ...
10  11  Finance   एक अर्ब रुपैयाँभन्दा बढी चुक्ता पुँजी भएका कम...
11  12  Finance  भारतको सबैभन्दा ठूलो वायुसेवा कम्पनी इन्डिगोले...
12  13  Finance  सरकारले निजी क्षेत्रलाई लगानीको वातावरण तयार प...
13  14  Finance  काठमाडौं तराई/मधेश द्रुतमार्ग (फास्ट ट्रयाक) ...
14  15  Finance  नेपाललाई मेला, सभा/सम्मेलन तथा विवाह गन्तव्यका...
15  16  Finance  लुम्बिनीमा ९ लाख ७४ हजार ३ सय ८१ हेक्टर वन क्ष...
16  17   Sports  एसियाली खेलकुदमा ई-स्पोर्ट्सले पहिलोपल्ट प्रवे...
17  18   Sports  चीनले फेरि एकपल्ट आफ्नो भूमिमा हुने १९ औं एसिय...
```

```
18  19   Sports  इजरायलको महिला फुटबल लिगमा हापोएल रानानाबाट खे...
19  20   Sports  पुलिसकी शुभाङ्गी श्रेष्ठले ११ औं कोरियन एम्बास...



==================================
Task 1: 2023-09-02 02:03:00.053374
==================================
```

Task 2.1: Frequency Analysis

Perform the frequency analysis on the text collected

```
# Task 2.1:Frequency Analysis
import matplotlib.pyplot as plt
import nltk
from nltk import FreqDist, word_tokenize

nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

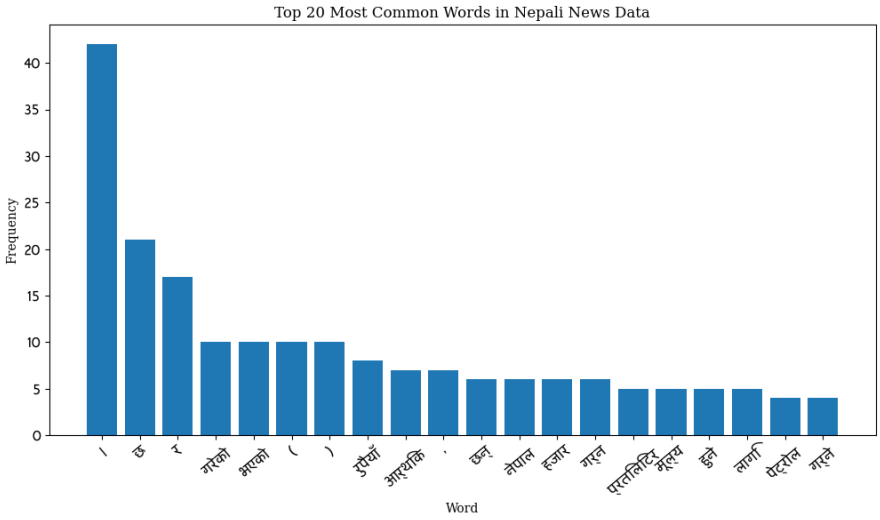
```
#Tokenize the text
tokens = df['News'].apply(nltk.word_tokenize)

# Flatten the list of tokens
all_tokens = [token for sublist in tokens for token in sublist]

print(len(all_tokens))
```

```
864
```

```
# Calculate word frequencies
freq_dist = FreqDist(all_tokens)

# Print the top 20 most common words
print(freq_dist.most_common(20))
# Get the most common 20 words and their frequencies
n = 20
common_words = freq_dist.most_common(n)
words, frequencies = zip(*common_words)
```

```
[('।', 42), ('छ', 21), ('र', 17), ('गरेको', 10), ('भएको', 10), ('(', 10), (')', 10), ('रुपैयाँ', 8), ('आर्थिक', 7), (',', 7), ('छन्'
```

```
import matplotlib.font_manager as font_manager

nep_font_prop = font_manager.FontProperties(fname="/content/Assignment/Mangal Regular.otf", size=12)
eng_font_properties = font_manager.FontProperties(family='serif', size=12)
plt.rcParams['font.family'] = 'serif'

#plot word frequency distribution
plt.figure(figsize=(12,6))
plt.bar(words,frequencies)
plt.title('Top 20 Most Common Words in Nepali News Data')
plt.xticks(rotation=40)
plt.xlabel('Word')
plt.ylabel('Frequency')
plt.xticks(fontproperties=nep_font_prop)
plt.yticks(fontproperties=nep_font_prop)
plt.show()
```




```
done('Task 2.1')
```

```
==================================
Task 2.1: 2023-09-02 02:03:00.077236
==================================
```

Task 2.2: Filter Stop words

Improve Performance analysis by filtering stop words (you can also develop rule based)

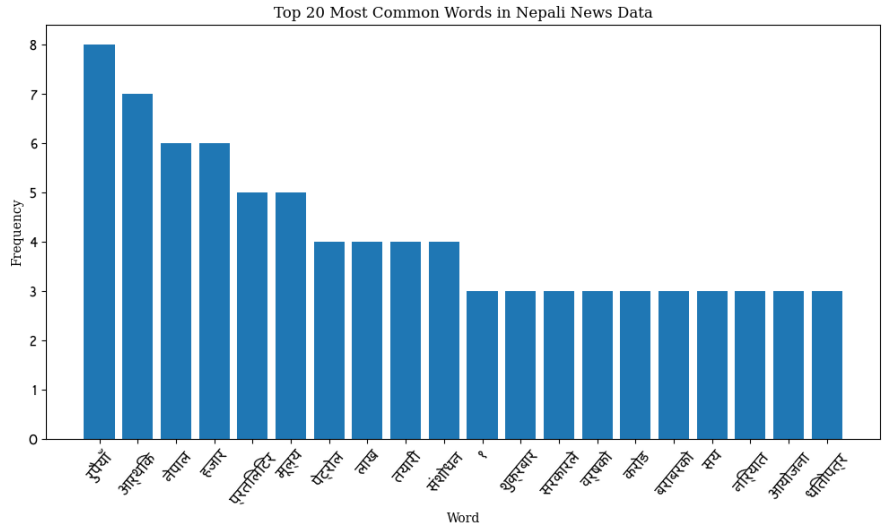
```
nltk.download('stopwords')
from nltk.corpus import stopwords

# Load the list of Nepali stop words
nepali_stop_words = set(stopwords.words('nepali'))
print(nepali_stop_words)
other_characters = ['।',',','|','-','‘’','()','(',')', 'छ।','र','पनि','छन्', 'गरेको','भएको','गरिएको','हो।','गर्न', 'गर्ने', 'एउटा','‘','रहेको
```

```
{'आजको', 'आफ्नै', 'लाई', 'चाहन्छु', 'हुन', 'भन्छु', 'भित्र', 'वास्तवमा', 'तिनी', 'भित्री', 'आत्म', 'हुन्छ', 'यो', 'जसबाट', 'पहिले', 'बरु', 'देखे
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
filter_words = list(nepali_stop_words) + list(other_characters)
print(filter_words)

tokens = []
for token in all_tokens:
    if token not in filter_words:
        tokens.append(token)
print(len(tokens))


# Calculate word frequencies
freq_dist = FreqDist(tokens)

# Print the top 20 most common words
print(freq_dist.most_common(20))
# Get the most common 20 words and their frequencies
n = 20
common_words = freq_dist.most_common(n)
words, frequencies = zip(*common_words)
```

```
['आजको', 'आफ्नै', 'लाई', 'चाहन्छु', 'हुन', 'भन्छु', 'भित्र', 'वास्तवमा', 'तिनी', 'भित्री', 'आत्म', 'हुन्छ', 'यो', 'जसबाट', 'पहिले', 'बरु', 'देखे
633
[('रुपैयाँ', 8), ('आर्थिक', 7), ('नेपाल', 6), ('हजार', 6), ('प्रतिलिटर', 5), ('मूल्य', 5), ('पेट्रोल', 4), ('लाख', 4), ('तयारी', 4), ('संशोधन
```

```
#plot word frequency distribution
plt.figure(figsize=(12,6))
plt.bar(words,frequencies)
plt.title('Top 20 Most Common Words in Nepali News Data')
plt.xticks(rotation=50)
plt.xlabel('Word')
plt.ylabel('Frequency')
plt.xticks(fontproperties=nep_font_prop)
plt.yticks(fontproperties=nep_font_prop)
plt.show()
```




```
done('Task 2.2')
```

```
==================================
Task 2.2: 2023-09-02 02:03:00.100074
==================================
```

Task 3: BoW

Task 3: BoW: Prepare Bag of Words (BoW) from the dataset

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer


# Create a CountVectorizer object
vectorizer = CountVectorizer()

# Fit and transform the text data to create the BoW representation
X_bow = vectorizer.fit_transform(df['News'])

# Convert the BoW representation to a DataFrame with feature names
bow_df = pd.DataFrame(X_bow.toarray(), columns=vectorizer.get_feature_names_out())

# Now, bow_df contains your Bag of Words representation
bow_df.head()
```

	अघ	अच	अझ	अत	अध	अन	अब	अभ	अर	अवस	...	५९	६५	७०	७२	७३	७४
0	0	0	0	0	0	0	1	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
2	0	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	1	0	0	1	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0

5 rows × 292 columns

```
done('Task 3')
```

```
==================================
Task 3: 2023-09-02 02:03:00.121642
==================================
```

Task 4: Classification

Classify the news based on Keywords and BoW you computed in Task 3.1

```
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(bow_df, df['Catogory'], test_size=0.3, random_state=20)

# Create a Naive Bayes classifier
clf = MultinomialNB()

# Train the classifier on the training data
clf.fit(X_train, y_train)

# Predict the categories for the test data
y_pred = clf.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
        Accuracy: 0.8333333333333334


    done('Task 4')

        ==================================
        Task 4: 2023-09-02 02:03:00.139045
        ==================================
```

✓ 0s completed at 9:33 PM