NLP Text Preprocessing Test

Total Marks: 50

Story

In the rugged terrain of the Western Ghats, Shivaji Maharaj, the founder of the Maratha Empire, was preparing for another day of strategic planning and military campaigns. Known for his innovative tactics and leadership, Shivaji Maharaj was determined to strengthen his kingdom and protect his people from external threats.

Shivaji Maharaj’s stronghold was the Raigad Fort, a formidable fortress perched high on a hilltop. The fort's strategic location allowed him to oversee the surrounding lands and defend against invaders. His trusted advisors and generals would gather in the fort's grand hall to discuss plans, share intelligence, and make critical decisions for the empire's expansion.

The evenings in Shivaji Maharaj's court were vibrant with activity. He would host feasts and gatherings to bolster the morale of his soldiers and celebrate victories. During these events, Shivaji Maharaj would address his people, sharing his vision for a prosperous and united Maratha Empire. His leadership inspired loyalty and courage among his subjects.

As night fell, Shivaji Maharaj would retire to his quarters, reflecting on the day's achievements and preparing for the challenges ahead. His dreams were filled with visions of a strong and resilient empire, where justice and valor would prevail.

Questions

1. **Tokenization (3 marks)**
   a. Tokenize the story into sentences.
   b. Tokenize the story into words.
2. **Lowercasing (3 marks)**
   Convert all words in the story to lowercase. Provide the transformed text.
3. **Stopword Removal (3 marks)**
   a. Remove common stopwords (e.g., 'the', 'was', 'a') from the story.
   b. List the removed stopwords.
4. **Punctuation Removal (3 marks)**
   Remove all punctuation marks from the story. Provide the cleaned text.
5. **Stemming (3 marks)**
   a. Apply stemming to the words in the story.
   b. Provide a few examples of stemmed words.
6. **Lemmatization (3 marks)**
   a. Apply lemmatization to the words in the story.
   b. Provide a few examples of lemmatized words.
7. **Named Entity Recognition (NER) (3 marks)**
   Identify and list all named entities (e.g., people, locations) in the story.
8. **Part-of-Speech Tagging (3 marks)**
   Tag each word in the story with its corresponding part of speech. Provide the tagged text for a few sentences.
9. **Term Frequency (2 marks)**
   Calculate the term frequency (TF) of the word "Shivaji" in the story.
10. **Frequency Distribution (3 marks)**
    Plot the frequency distribution of the top 10 most common words in the story. Describe the results.
11. **Token Count (2 marks)**
    Count the total number of tokens (words) in the story.
12. **Sentence Count (2 marks)**
    Count the total number of sentences in the story.
13. **Stopword Frequency (2 marks)**
    Calculate the frequency of stopwords in the story. Provide the top 3 most frequent stopwords.
14. **Longest Word (2 marks)**
    Identify the longest word in the story and its length.

15. **Word Cloud (3 marks)**
    Create a word cloud based on the story. Describe the most prominent words in the cloud.
16. **Noun Identification (2 marks)**
    Identify and list all nouns in the story.
17. **Verb Identification (2 marks)**
    Identify and list all verbs in the story.
18. **Word Context (2 marks)**
    Find and list the sentences where the word "empire" appears.
19. **Named Entity Categorization (2 marks)**
    Categorize the identified named entities into people, locations, and organizations.
20. **Word Frequency Comparison (2 marks)**
    Compare the frequency of the words "kingdom" and "feast" in the story. Provide their counts and analyze the difference.

Instructions

- Use Python libraries like `nltk`, `spaCy`, `wordcloud`, etc., where appropriate.
- Ensure to follow the specified format for each task.
- Write clear and concise explanations for your results where required.
- Submit your notebook as a PDF file.

Good luck!

```
import nltk
import string
from nltk.corpus import stopwords
import matplotlib.pyplot as plt
from nltk import pos_tag
from nltk.stem import wordnet
from nltk import ne_chunk
from nltk.probability import FreqDist
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize
import spacy
from collections import Counter
from wordcloud import WordCloud
import matplotlib.pyplot as plt
nltk.download('punkt')
nltk.download('words')
nltk.download('wordnet')
nltk.download('stopwords')
nltk.download('maxent_ne_chunker')
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package words to /root/nltk_data...
[nltk_data]   Package words is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package maxent_ne_chunker is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
True
```

```
text = """In the rugged terrain of the Western Ghats, Shivaji Maharaj, the founder of the Maratha Empire, was preparing for another day of strategic planning and military car

Shivaji Maharaj’s stronghold was the Raigad Fort, a formidable fortress perched high on a hilltop. The fort's strategic location allowed him to oversee the surrounding lands ar

The evenings in Shivaji Maharaj's court were vibrant with activity. He would host feasts and gatherings to bolster the morale of his soldiers and celebrate victories. During the

As night fell, Shivaji Maharaj would retire to his quarters, reflecting on the day's achievements and preparing for the challenges ahead. His dreams were filled with visions of a
```

```
#1
sent = sent_tokenize(text)
print(sent)
```

```
['In the rugged terrain of the Western Ghats, Shivaji Maharaj, the founder of the Maratha Empire, was preparing for another day of strategic planning and military camp
```

```
words = word_tokenize(text)
print(words)
```

```
['In', 'the', 'rugged', 'terrain', 'of', 'the', 'Western', 'Ghats', ',', 'Shivaji', 'Maharaj', ',', 'the', 'founder', 'of', 'the', 'Maratha', 'Empire', ',', 'was', 'preparing', 'for', 'another', 'da
```

```
#2
def lowercase_text(text):
    return text.lower()

lowercase_text(text)
```

```
'in the rugged terrain of the western ghats, shivaji maharaj, the founder of the maratha empire, was prepar
ing for another day of strategic planning and military campaigns. known for his innovative tactics and leade
rship, shivaji maharaj was determined to strengthen his kingdom and protect his people from external thre
ats.\n\nshivaji maharaj’s stronghold was the raigad fort, a formidable fortress perched high on a hilltop. the
fort's strategic location allowed him to oversee the surrounding lands and defend against invaders. his trust
```

```
#next line removal
text = text.replace("\n"," ")


#3
def remove_stopwords(text):
  stop_words = set(stopwords.words("english"))
  word_tokens = word_tokenize(text)
  filtered_text = [word for word in word_tokens if word not in string.punctuation]
  rem_text =  [word for word in filtered_text if word  in stop_words]
  filtered_text = [word for word in filtered_text if word not in stop_words]

  return filtered_text,rem_text


filtered_text,rem_text = remove_stopwords(text)
print(filtered_text)
print("\nBelow is the removed stopwords\n")
print(rem_text)
```

```
['In', 'rugged', 'terrain', 'Western', 'Ghats', 'Shivaji', 'Maharaj', 'founder', 'Maratha', 'Empire', 'preparing', 'another', 'day', 'strategic', 'planning', 'military', 'campaigns', 'Kn

Below is the removed stopwords

['the', 'of', 'the', 'the', 'of', 'the', 'was', 'for', 'of', 'and', 'for', 'his', 'and', 'was', 'to', 'his', 'and', 'his', 'from', 's', 'was', 'the', 'a', 'on', 'a', 'him', 'to', 'the', 'and', 'against', 'and',
```

```
#4
def rem_punct(text):
    translator = str.maketrans('', '', string.punctuation)
    return text.translate(translator)

rem_punct(text)
```

```
'In the rugged terrain of the Western Ghats Shivaji Maharaj the founder of the Maratha Empire was prepari
ng for another day of strategic planning and military campaigns Known for his innovative tactics and leaders
hip Shivaji Maharaj was determined to strengthen his kingdom and protect his people from external threats
Shivaji Maharaj’s stronghold was the Raigad Fort a formidable fortress perched high on a hilltop The forts st
rategic location allowed him to oversee the surrounding lands and defend against invaders His trusted advis
ors and generals would gather in the forts grand hall to discuss plans share intelligence and make critical de
```

```
#5
stemmer = PorterStemmer()

def stem_words(text):
  word_tokens = word_tokenize(text)
  stems = [stemmer.stem(word) for word in word_tokens]
  return stems


stemmed_words=stem_words(text)
print(stemmed_words)
```

```
['in', 'the', 'rug', 'terrain', 'of', 'the', 'western', 'ghat', ',', 'shivaji', 'maharaj', ',', 'the', 'founder', 'of', 'the', 'maratha', 'empir', ',', 'wa', 'prepar', 'for', 'anoth', 'day', 'of', 'strat
```

```
#6
def lemmatize_word(text):
  word_tokens = word_tokenize(text)
  lemmas = [wordnet.WordNetLemmatizer().lemmatize(word, pos='v') for word in word_tokens]
  return lemmas


lemmatized_words = lemmatize_word(text)
print(lemmatized_words)
```

```
['In', 'the', 'rugged', 'terrain', 'of', 'the', 'Western', 'Ghats', ',', 'Shivaji', 'Maharaj', ',', 'the', 'founder', 'of', 'the', 'Maratha', 'Empire', ',', 'be', 'prepare', 'for', 'another', 'day',
```

```
#7
def ner(text):
  word_tokens = word_tokenize(text)
  word_pos = pos_tag(word_tokens)
  print(ne_chunk(word_pos))


ner(text)
```

```
          prevail/VB
          ./.)
```

```
#8
def pos_tagg(text):
  word_tokens = word_tokenize(text)
  return pos_tag(word_tokens)
```

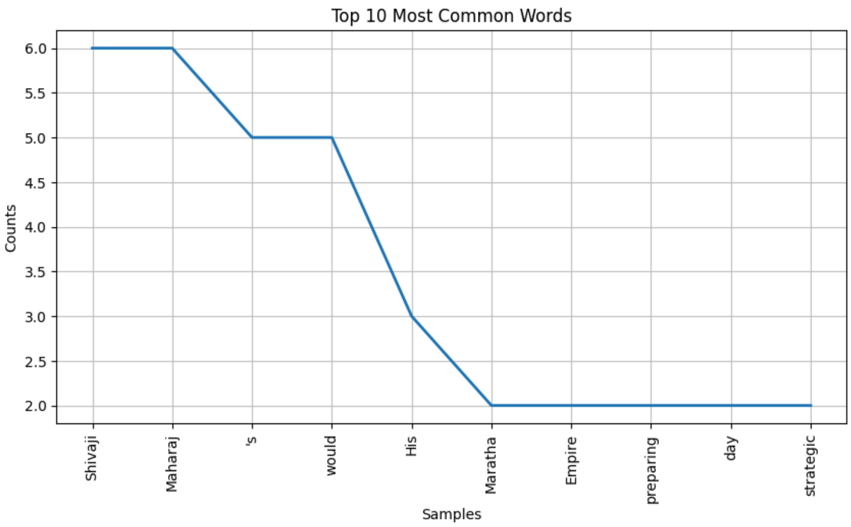
```
pos_tagg(text)
```

```
 ('Maratha', 'NNP'),
 ('Empire', 'NNP'),
 ('.', '.'),
 ('His', 'PRP$'),
 ('leadership', 'NN'),
 ('inspired', 'VBD'),
 ('loyalty', 'NN'),
 ('and', 'CC'),
 ('courage', 'NN'),
 ('among', 'IN'),
 ('his', 'PRP$'),
 ('subjects', 'NNS'),
 ('.', '.'),
 ('As', 'IN'),
 ('night', 'NN'),
 ('fell', 'VBD'),
 (',', ','),
 ('Shivaji', 'NNP'),
 ('Maharaj', 'NNP'),
 ('would', 'MD'),
 ('retire', 'VB'),
 ('to', 'TO'),
 ('his', 'PRP$'),
 ('quarters', 'NNS'),
 (',', ','),
 ('reflecting', 'VBG'),
 ('on', 'IN'),
 ('the', 'DT'),
 ('day', 'NN'),
 ("'s", 'POS'),
 ('achievements', 'NNS'),
 ('and', 'CC'),
 ('preparing', 'VBG'),
 ('for', 'IN'),
 ('the', 'DT'),
 ('challenges', 'NNS'),
 ('ahead', 'RB'),
 ('.', '.'),
 ('His', 'PRP$'),
 ('dreams', 'NNS'),
 ('were', 'VBD'),
 ('filled', 'VBN'),
 ('with', 'IN'),
 ('visions', 'NNS'),
 ('of', 'IN'),
 ('a', 'DT'),
 ('strong', 'JJ'),
 ('and', 'CC'),
 ('resilient', 'JJ'),
 ('empire', 'NN'),
 (',', ','),
 ('where', 'WRB'),
 ('justice', 'NN'),
 ('and', 'CC'),
 ('valor', 'NN'),
 ('would', 'MD'),
 ('prevail', 'VB'),
 ('.', '.')]
```

```
#9
tf_shivaji = stemmed_words.count('shivaji')
print("Term Frequency of 'Shivaji':", tf_shivaji)
```

```
Term Frequency of 'Shivaji': 6
```

```
#10
fdist = FreqDist(filtered_text)
plt.figure(figsize=(10, 5))
fdist.plot(10, title='Top 10 Most Common Words')
plt.show()
```



```
#11
token_count = len(words)
print("Total number of tokens:", token_count)
```

```
Total number of tokens: 227
```

```
#12
sentence_count = len(sent)
print("Total number of sentences:", sentence_count)
```

```
Total number of sentences: 11
```

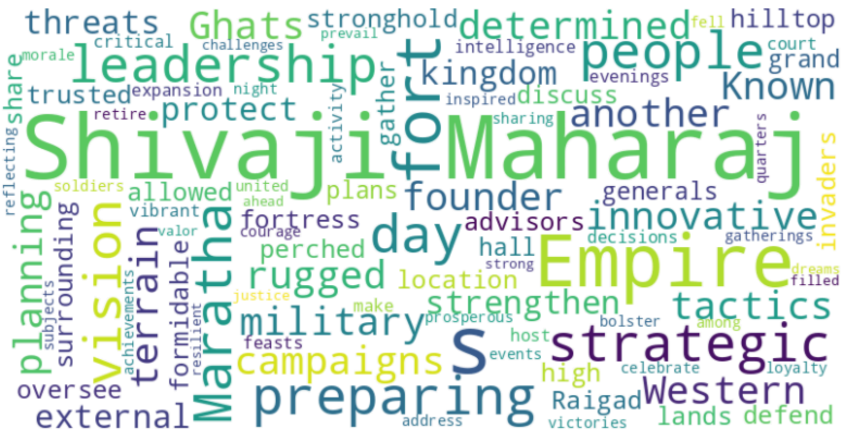
```
#13
stopword_counts = Counter(rem_text)
top_stopwords = stopword_counts.most_common(3)
print("Top 3 most frequent stopwords:", top_stopwords)
```

```
Top 3 most frequent stopwords: [('and', 13), ('the', 11), ('his', 8)]
```

```
#14
longest_word = max(filtered_text, key=len)
print(longest_word, len(longest_word))
```

```
intelligence 12
```

```
#15
filtered_words= ' '.join(filtered_text)
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(filtered_words)
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



```
#16
nlp = spacy.load('en_core_web_sm')
doc = nlp(text)

nouns = [token.text for token in doc if token.pos_ == 'NOUN']
print("Nouns in the story:", nouns)
```

```
Nouns in the story: ['terrain', 'founder', 'day', 'planning', 'campaigns', 'tactics', 'leadership', 'kingdom', 'people', 'threats', 'stronghold', 'fortress', 'hilltop', 'fort', 'location',
```

```
#17
verbs = [token.text for token in doc if token.pos_ == 'VERB']
print("Verbs in the story:", verbs)
```

```
Verbs in the story: ['preparing', 'Known', 'strengthen', 'protect', 'perched', 'allowed', 'oversee', 'surrounding', 'defend', 'trusted', 'gather', 'discuss', 'make', 'host', 'bolster
```

```
#18
empire_sentences = [s for s in sent if 'empire' in s.lower()]
print(empire_sentences)
```

```
['In the rugged terrain of the Western Ghats, Shivaji Maharaj, the founder of the Maratha Empire, was preparing for another day of strategic planning and military camp
```

```
#19
people = []
locations = []
organizations = []


for ent in doc.ents:
    if ent.label_ == 'PERSON':
        people.append(ent.text)
    elif ent.label_ == 'LOC' :
        locations.append(ent.text)
    elif ent.label_ == 'ORG':
        organizations.append(ent.text)


print("People:", people)
print("Locations:", locations)
print("Organizations:", organizations)
```

```
People: ['Shivaji Maharaj', 'Shivaji Maharaj', 'Shivaji Maharaj’s', "Shivaji Maharaj's", 'Shivaji Maharaj', 'Maratha Empire', 'Shivaji Maharaj']
Locations: []
Organizations: []
```

```
#20
freq_kingdom = filtered_text.count('kingdom')
freq_feast = filtered_text.count('feast')

print("Frequency of 'kingdom':", freq_kingdom)
print("Frequency of 'feast':", freq_feast)
```

```
Frequency of 'kingdom': 1
Frequency of 'feast': 0
```