GUJJA RAJU 2403A52018 CSE AIML BATCH-02

```
import nltk
import spacy
import matplotlib.pyplot as plt
import pandas as pd
nltk.download('punkt')
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 averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
True
```

```
essay_text = """The rapid advancement of artificial intelligence (AI) presents

Ethical concerns surrounding AI development include algorithmic bias, privacy i

The future of human-computer interaction is poised for transformation. AI-power
"""
print("Essay text loaded successfully into 'essay_text' variable.")
```

```
Essay text loaded successfully into 'essay_text' variable.
```

```
nltk.download('punkt_tab')
nltk_tokens = nltk.word_tokenize(essay_text)
print(f"NLTK tokens (first 10): {nltk_tokens[:10]}")
print(f"Total NLTK tokens: {len(nltk_tokens)}")
```

```
NLTK tokens (first 10): ['The', 'rapid', 'advancement', 'of', 'artificial', 'int
Total NLTK tokens: 258
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
```

```
nltk.download('averaged_perceptron_tagger_eng')
nltk_pos_tags = nltk.pos_tag(nltk_tokens)
print(f"NLTK POS tags (first 10): {nltk_pos_tags[:10]}")
print(f"Total NLTK POS tags: {len(nltk_pos_tags)}")
```

```
NLTK POS tags (first 10): [('The', 'DT'), ('rapid', 'JJ'), ('advancement', 'NN')
Total NLTK POS tags: 258
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger_eng is already up-to-
[nltk_data]       date!
```

```
nlp = spacy.load("en_core_web_sm")
spacy_doc = nlp(essay_text)
spacy_tokens = [token.text for token in spacy_doc]
print(f"spaCy tokens (first 10): {spacy_tokens[:10]}")
print(f"Total spaCy tokens: {len(spacy_tokens)}")
```

```
spaCy tokens (first 10): ['The', 'rapid', 'advancement', 'of', 'artificial', 'in
Total spaCy tokens: 277
```

```
spacy_pos_tags = [(token.text, token.pos_) for token in spacy_doc]
print(f"spaCy POS tags (first 10): {spacy_pos_tags[:10]}")
print(f"Total spaCy POS tags: {len(spacy_pos_tags)}")
```

```
spaCy POS tags (first 10): [('The', 'DET'), ('rapid', 'ADJ'), ('advancement', 'N
Total spaCy POS tags: 277
```

```
nltk_unique_tags = sorted(list(set([tag for word, tag in nltk_pos_tags])))
print(f"Unique NLTK POS Tags: {nltk_unique_tags}")
```

```
Unique NLTK POS Tags: ['(', ')', ',', '.', 'CC', 'DT', 'IN', 'JJ', 'MD', 'NN', '
```

```
spacy_unique_tags = sorted(list(set([tag for word, tag in spacy_pos_tags])))
print(f"Unique spaCy POS Tags: {spacy_unique_tags}")
```

```
Unique spaCy POS Tags: ['ADJ', 'ADP', 'ADV', 'AUX', 'CCONJ', 'DET', 'NOUN', 'PAR
```

```
pos_tag_mapping = {
    'NLTK_NOUN_TAGS': ['NN', 'NNS', 'NNP', 'NNPS'],
    'SPACY_NOUN_TAGS': ['NOUN', 'PROPN'],
    'NLTK_VERB_TAGS': ['VB', 'VBD', 'VBG', 'VBN', 'VBP', 'VBZ'],
    'SPACY_VERB_TAGS': ['VERB', 'AUX'],
    'NLTK_ADJECTIVE_TAGS': ['JJ', 'JJR', 'JJS'],
    'SPACY_ADJECTIVE_TAGS': ['ADJ'],
    'NLTK_ADVERB_TAGS': ['RB', 'RBR', 'RBS'],
    'SPACY_ADVERB_TAGS': ['ADV'],
    'NLTK_PUNCTUATION_TAGS': [',', '.', ':', '(', ')', '``', "''"],
    'SPACY_PUNCTUATION_TAGS': ['PUNCT']
}
print("POS Tag Mapping:")
for key, value in pos_tag_mapping.items():
    print(f"  {key}: {value}")
```

```
POS Tag Mapping:
  NLTK_NOUN_TAGS: ['NN', 'NNS', 'NNP', 'NNPS']
  SPACY_NOUN_TAGS: ['NOUN', 'PROPN']
  NLTK_VERB_TAGS: ['VB', 'VBD', 'VBG', 'VBN', 'VBP', 'VBZ']
  SPACY_VERB_TAGS: ['VERB', 'AUX']
  NLTK_ADJECTIVE_TAGS: ['JJ', 'JJR', 'JJS']
  SPACY_ADJECTIVE_TAGS: ['ADJ']
  NLTK_ADVERB_TAGS: ['RB', 'RBR', 'RBS']
  SPACY_ADVERB_TAGS: ['ADV']
  NLTK_PUNCTUATION_TAGS: [',', '.', ':', '(', ')', '``', "''"]
  SPACY_PUNCTUATION_TAGS: ['PUNCT']
```

```
nltk_nouns = []
nltk_verbs = []
spacy_nouns = []
spacy_verbs = []

for word, tag in nltk_pos_tags:
    if tag in pos_tag_mapping['NLTK_NOUN_TAGS']:
        nltk_nouns.append(word.lower())
    elif tag in pos_tag_mapping['NLTK_VERB_TAGS']:
        nltk_verbs.append(word.lower())

for word, tag in spacy_pos_tags:
    if tag in pos_tag_mapping['SPACY_NOUN_TAGS']:
        spacy_nouns.append(word.lower())
    elif tag in pos_tag_mapping['SPACY_VERB_TAGS']:
        spacy_verbs.append(word.lower())

print(f"NLTK Nouns (first 10): {nltk_nouns[:10]}")
print(f"NLTK Verbs (first 10): {nltk_verbs[:10]}")
print(f"spaCy Nouns (first 10): {spacy_nouns[:10]}")
print(f"spaCy Verbs (first 10): {spacy_verbs[:10]}")
```

```
NLTK Nouns (first 10): ['advancement', 'intelligence', 'ai', 'opportunities', 'c
NLTK Verbs (first 10): ['presents', 'explore', 'argue', 'lead', 'contend', 'crea
spaCy Nouns (first 10): ['advancement', 'intelligence', 'ai', 'opportunities', '
spaCy Verbs (first 10): ['presents', 'will', 'explore', 'argue', 'will', 'lead',
```

```
from collections import Counter

print("Counter imported successfully.")
```

```
Counter imported successfully.
```

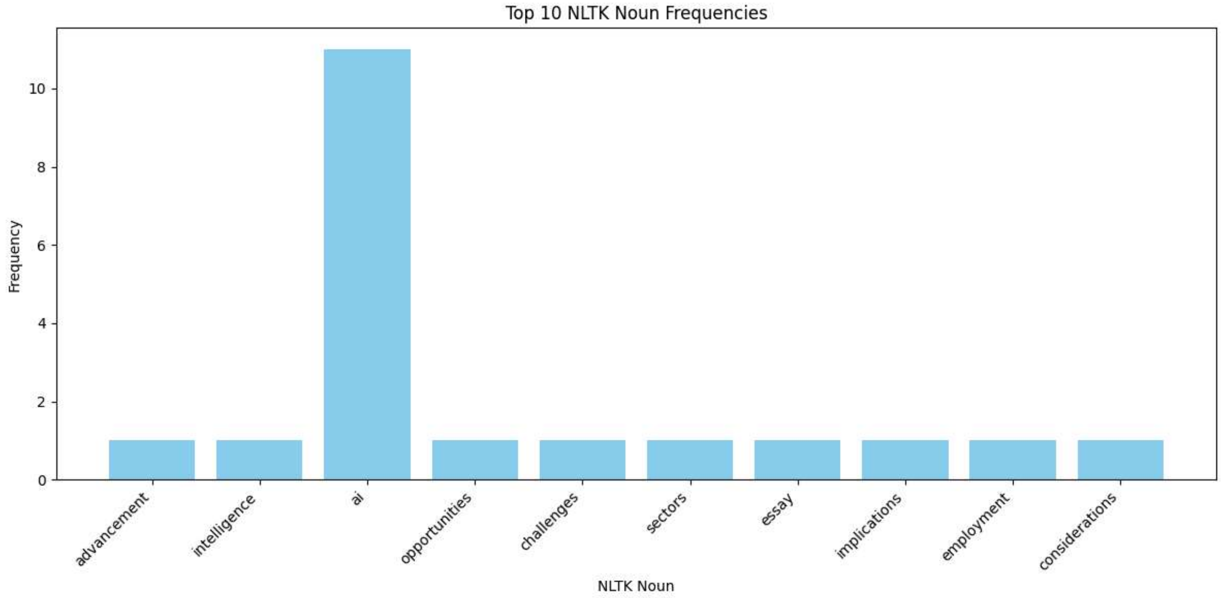
```
nltk_noun_freq = Counter(nltk_nouns)
nltk_verb_freq = Counter(nltk_verbs)
spacy_noun_freq = Counter(spacy_nouns)
spacy_verb_freq = Counter(spacy_verbs)

print(f"Most common NLTK Nouns: {nltk_noun_freq.most_common(10)}")
print(f"Most common NLTK Verbs: {nltk_verb_freq.most_common(10)}")
print(f"Most common spaCy Nouns: {spacy_noun_freq.most_common(10)}")
print(f"Most common spaCy Verbs: {spacy_verb_freq.most_common(10)}")
```

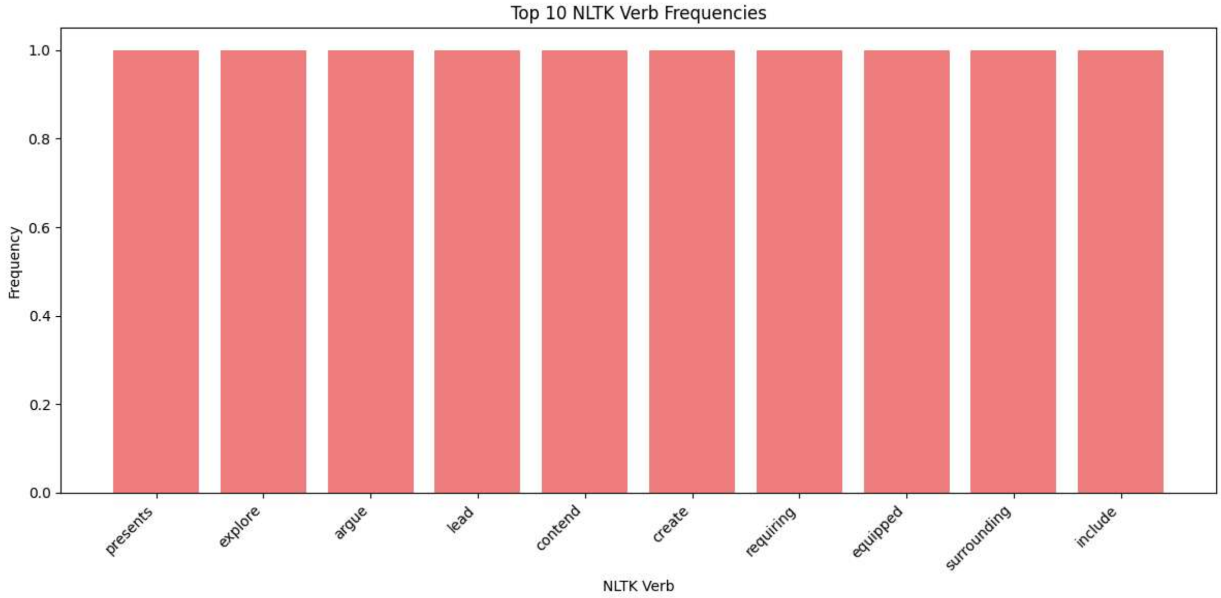
```
Most common NLTK Nouns: [('ai', 11), ('future', 3), ('systems', 3), ('interactic
Most common NLTK Verbs: [('is', 2), ('presents', 1), ('explore', 1), ('argue', 1
Most common spaCy Nouns: [('ai', 12), ('future', 3), ('systems', 3), ('computer'
Most common spaCy Verbs: [('will', 3), ('is', 2), ('presents', 1), ('explore', 1
```

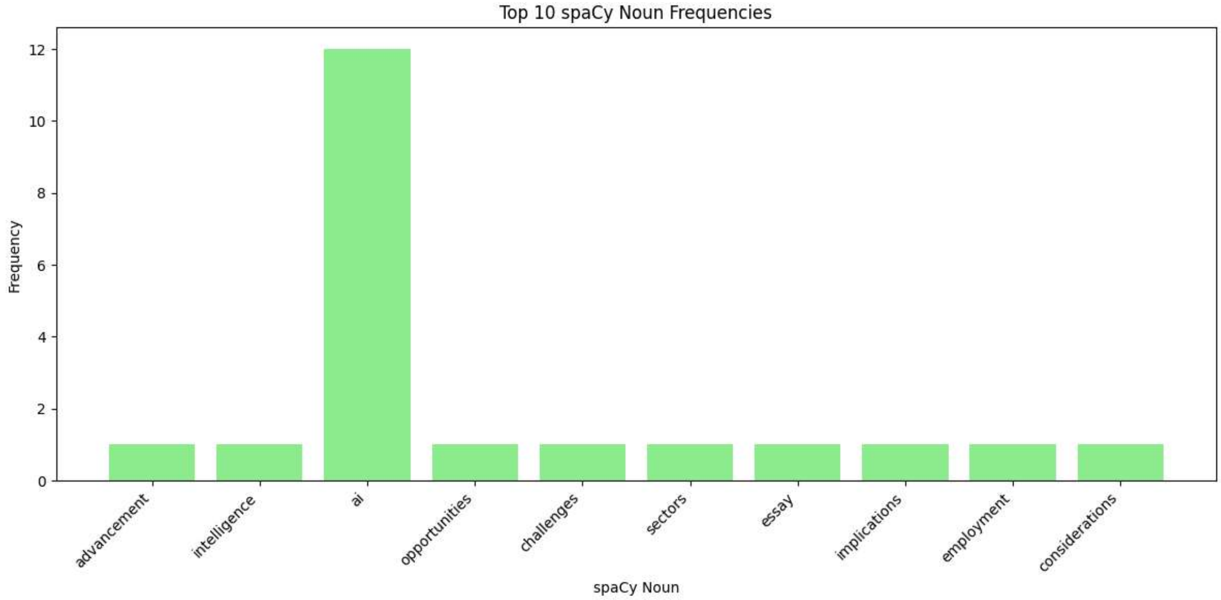
```
df_nltk_nouns = pd.DataFrame(nltk_noun_freq.items(), columns=['Noun', 'Frequenc
df_nltk_verbs = pd.DataFrame(nltk_verb_freq.items(), columns=['Verb', 'Frequenc
df_spacy_nouns = pd.DataFrame(spacy_noun_freq.items(), columns=['Noun', 'Freque
df_spacy_verbs = pd.DataFrame(spacy_verb_freq.items(), columns=['Verb', 'Freque

print("NLTK Nouns DataFrame (first 5 rows):")
print(df_nltk_nouns.head())
print("\nNLTK Verbs DataFrame (first 5 rows):")
print(df_nltk_verbs.head())
print("\nspaCy Nouns DataFrame (first 5 rows):")
print(df_spacy_nouns.head())
print("\nspaCy Verbs DataFrame (first 5 rows):")
print(df_spacy_verbs.head())
```

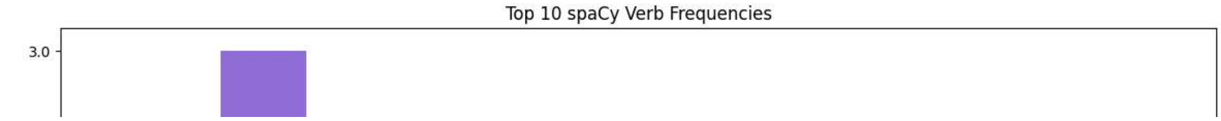
```
NLTK Nouns DataFrame (first 5 rows):
           Noun  Frequency
0   advancement          1
1  intelligence          1
2            ai         11
3 opportunities          1
4    challenges          1

NLTK Verbs DataFrame (first 5 rows):
       Verb  Frequency
0  presents          1
1   explore          1
2     argue          1
3      lead          1
4   contend          1

spaCy Nouns DataFrame (first 5 rows):
           Noun  Frequency
0   advancement          1
1  intelligence          1
2            ai         12
3 opportunities          1
4    challenges          1

spaCy Verbs DataFrame (first 5 rows):
       Verb  Frequency
0  presents          1
1      will          3
2   explore          1
3     argue          1
4      lead          1
```

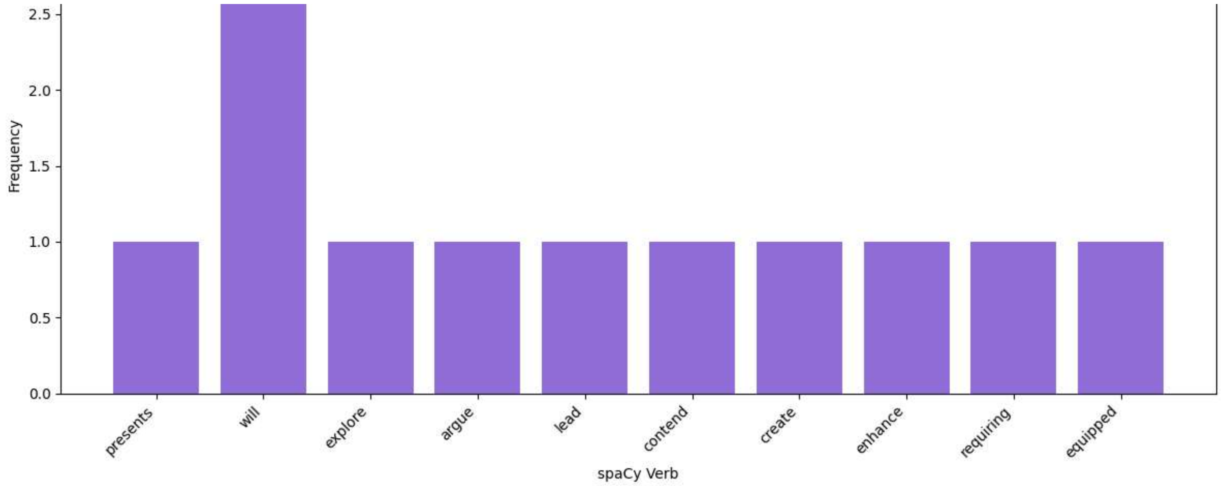
```
top_n = 10

# Plot NLTK Noun Frequencies
plt.figure(figsize=(12, 6))
plt.bar(df_nltk_nouns['Noun'].head(top_n), df_nltk_nouns['Frequency'].head(top_
plt.xlabel('NLTK Noun')
plt.ylabel('Frequency')
plt.title(f'Top {top_n} NLTK Noun Frequencies')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

# Plot NLTK Verb Frequencies
plt.figure(figsize=(12, 6))
plt.bar(df_nltk_verbs['Verb'].head(top_n), df_nltk_verbs['Frequency'].head(top_
plt.xlabel('NLTK Verb')
plt.ylabel('Frequency')
plt.title(f'Top {top_n} NLTK Verb Frequencies')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

# Plot spaCy Noun Frequencies
plt.figure(figsize=(12, 6))
plt.bar(df_spacy_nouns['Noun'].head(top_n), df_spacy_nouns['Frequency'].head(to
plt.xlabel('spaCy Noun')
plt.ylabel('Frequency')
plt.title(f'Top {top_n} spaCy Noun Frequencies')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

# Plot spaCy Verb Frequencies
plt.figure(figsize=(12, 6))
plt.bar(df_spacy_verbs['Verb'].head(top_n), df_spacy_verbs['Frequency'].head(to
plt.xlabel('spaCy Verb')
plt.ylabel('Frequency')
plt.title(f'Top {top_n} spaCy Verb Frequencies')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```


Top 10 NLTK Noun Frequencies
Frequency
0
2
4
6
8
10
advancement
intelligence
ai
opportunities
challenges
sectors
essay
implications
employment
considerations
NLTK Noun


Top 10 NLTK Verb Frequencies
Frequency
0.0
0.2
0.4
0.6
0.8
1.0
presents
explore
argue
lead
contend
create
requiring
equipped
surrounding
include
NLTK Verb


Top 10 spaCy Noun Frequencies
Frequency
0
2
4
6
8
10
12
advancement
intelligence
ai
opportunities
challenges
sectors
essay
implications
employment
considerations
spaCy Noun


Top 10 spaCy Verb Frequencies
3.0


2.5
2.0
1.5
1.0
0.5
0.0
Frequency
presents
will
explore
argue
lead
contend
create
enhance
requiring
equipped
spaCy Verb