

```

#importing libraries
import nltk
import spacy

#Downloading resources
# nltk.download('punkt')
# nltk.download('punkt_tab')
# nltk.download('averaged_perceptron_tagger')
# nltk.download('averaged_perceptron_tagger_eng')

# Datasets

essay_text = """
Artificial intelligence has become a significant area of research in
computer science.
It enables machines to perform tasks that traditionally require human
intelligence.
Academic institutions are increasingly focusing on AI-driven solutions
to improve efficiency.
The development of intelligent systems depends on data, algorithms,
and computational power.
"""

#Tokenization
from nltk.tokenize import word_tokenize
tokens = word_tokenize(essay_text)
tokens

#POS tagging using nltk
nltk_pos_tags = nltk.pos_tag(tokens)

nltk_clean = [
    (word, tag)
    for word, tag in nltk_pos_tags
    if word.isalpha()
]

print("\nPOS tags using nltk: \n")
for word, tag in nltk_pos_tags:
    print(f"{word} : {tag}")

#POS tagging using spacy
nlp = spacy.load("en_core_web_sm")
doc = nlp(essay_text)

spacy_pos_tags = spacy_pos_tags = [
    (token.text, token.pos_)
    for token in doc
    if not token.is_space and token.pos_ != "PUNCT"
]

```

```

spacy_clean = [
    (token.text, token.pos_)
    for token in doc
    if token.is_alpha
]

print("\nPOS tags using spacy: \n")
for word, tag in spacy_pos_tags:
    print(f"{word} : {tag}")

#Comparing tag sets for academic vocabulary
comparison = []

for ((word_nltk, nltk_tag), (word_spacy, spacy_tag)) in
zip(nltk_clean, spacy_clean):
    comparison.append((word_nltk, nltk_tag, spacy_tag))

print("\nComparison (NLTK vs spaCy):\n")
for word, nltk_tag, spacy_tag in comparison:
    print(f"{word:15} {nltk_tag:5} → {spacy_tag}")

#Extracting nouns (concepts) and verbs (arguments) to analyze frequency
nouns = [
    token.text.lower()
    for token in doc
    if token.pos_ == "NOUN"
]

verbs = [
    token.text.lower()
    for token in doc
    if token.pos_ == "VERB"
]

from collections import Counter

noun_freq = Counter(nouns)
verb_freq = Counter(verbs)

print("\nNoun Frequency (Concepts):\n")
for word, count in noun_freq.most_common():
    print(f"{word:15} : {count}")

print("\nVerb Frequency (Arguments):\n")
for word, count in verb_freq.most_common():
    print(f"{word:15} : {count}")

```

POS tags using nltk:

Artificial : JJ  
intelligence : NN  
has : VBZ  
become : VBN  
a : DT  
significant : JJ  
area : NN  
of : IN  
research : NN  
in : IN  
computer : NN  
science : NN  
. : .  
It : PRP  
enables : VBZ  
machines : NNS  
to : TO  
perform : VB  
tasks : NNS  
that : WDT  
traditionally : RB  
require : VBP  
human : JJ  
intelligence : NN  
. : .  
Academic : JJ  
institutions : NNS  
are : VBP  
increasingly : RB  
focusing : VBG  
on : IN  
AI-driven : NNP  
solutions : NNS  
to : TO  
improve : VB  
efficiency : NN  
. : .  
The : DT  
development : NN  
of : IN  
intelligent : JJ  
systems : NNS  
depends : VBZ  
on : IN  
data : NNS  
, : ,  
algorithms : NN

, : ,  
and : CC  
computational : JJ  
power : NN  
. : .

POS tags using spacy:

Artificial : ADJ  
intelligence : NOUN  
has : AUX  
become : VERB  
a : DET  
significant : ADJ  
area : NOUN  
of : ADP  
research : NOUN  
in : ADP  
computer : NOUN  
science : NOUN  
It : PRON  
enables : VERB  
machines : NOUN  
to : PART  
perform : VERB  
tasks : NOUN  
that : PRON  
traditionally : ADV  
require : VERB  
human : ADJ  
intelligence : NOUN  
Academic : ADJ  
institutions : NOUN  
are : AUX  
increasingly : ADV  
focusing : VERB  
on : ADP  
AI : PROPN  
driven : VERB  
solutions : NOUN  
to : PART  
improve : VERB  
efficiency : NOUN  
The : DET  
development : NOUN  
of : ADP  
intelligent : ADJ  
systems : NOUN  
depends : VERB  
on : ADP

data : NOUN  
algorithms : NOUN  
and : CCONJ  
computational : ADJ  
power : NOUN

#### Comparison (NLTK vs spaCy):

Artificial	JJ	→ ADJ
intelligence	NN	→ NOUN
has	VBZ	→ AUX
become	VBN	→ VERB
a	DT	→ DET
significant	JJ	→ ADJ
area	NN	→ NOUN
of	IN	→ ADP
research	NN	→ NOUN
in	IN	→ ADP
computer	NN	→ NOUN
science	NN	→ NOUN
It	PRP	→ PRON
enables	VBZ	→ VERB
machines	NNS	→ NOUN
to	TO	→ PART
perform	VB	→ VERB
tasks	NNS	→ NOUN
that	WDT	→ PRON
traditionally	RB	→ ADV
require	VBP	→ VERB
human	JJ	→ ADJ
intelligence	NN	→ NOUN
Academic	JJ	→ ADJ
institutions	NNS	→ NOUN
are	VBP	→ AUX
increasingly	RB	→ ADV
focusing	VBG	→ VERB
on	IN	→ ADP
solutions	NNS	→ PROPN
to	TO	→ VERB
improve	VB	→ NOUN
efficiency	NN	→ PART
The	DT	→ VERB
development	NN	→ NOUN
of	IN	→ DET
intelligent	JJ	→ NOUN
systems	NNS	→ ADP
depends	VBZ	→ ADJ
on	IN	→ NOUN
data	NNS	→ VERB
algorithms	NN	→ ADP

and	CC	→ NOUN
computational	JJ	→ NOUN
power	NN	→ CCONJ

#### Noun Frequency (Concepts):

intelligence	: 2
area	: 1
research	: 1
computer	: 1
science	: 1
machines	: 1
tasks	: 1
institutions	: 1
solutions	: 1
efficiency	: 1
development	: 1
systems	: 1
data	: 1
algorithms	: 1
power	: 1

#### Verb Frequency (Arguments):

become	: 1
enables	: 1
perform	: 1
require	: 1
focusing	: 1
driven	: 1
improve	: 1
depends	: 1