Text Preprocessing with NLTK and spaCy

Aim

POS tagging in academic writing to analyze structure and vocabulary.

Requirements

- Google Colab
- Python 3.x
- Libraries: nltk, spacy

```
!pip install nltk spacy
!python -m spacy download en_core_web_sm
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.12/dist-packages (3.9.1)
Requirement already satisfied: spacy in /usr/local/lib/python3.12/dist-packages (3.8.11)
Requirement already satisfied: click in /usr/local/lib/python3.12/dist-packages (from nltk) (8.3.1
Requirement already satisfied: joblib in /usr/local/lib/python3.12/dist-packages (from nltk) (1.5.
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.12/dist-packages (from nl
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from nltk) (4.67.1
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.12/dist-packa
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packa
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-package
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (fro
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (f
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (fro
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (fr
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (fro
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (fr
Requirement already satisfied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.12/dist-packages (from spac
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/d
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.1
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from spacy)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from sp
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (f
Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-package
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from reque
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from
Requirement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packag
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from ji
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart-open<8
Collecting en-core-web-sm==3.8.0
  Using cached https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-3.8.0/er
✓ Download and installation successful
You can now load the package via spacy.load('en_core_web_sm')
⚠ Restart to reload dependencies
If you are in a Jupyter or Colab notebook, you may need to restart Python in
order to load all the package's dependencies. You can do this by selecting the
'Restart kernel' or 'Restart runtime' option.
```

Import Libraries

```
import nltk
import spacy
from collections import Counter
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nlp = spacy.load('en_core_web_sm')
```

```
[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!
```

Sample Academic Essay Text

```
text = '''Artificial Intelligence has transformed modern research. Researchers analyze large data
text
```

```
'Artificial Intelligence has transformed modern research. Researchers analyze large datasets to i
mprove decision making and develop intelligent systems that solve complex problems.'
```

Tokenization using NLTK

```
nltk.download('punkt_tab')
tokens_nltk = nltk.word_tokenize(text)
tokens_nltk
```

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
['Artificial',
 'Intelligence',
 'has',
 'transformed',
 'modern',
 'research',
 '.',
 'Researchers',
 'analyze',
 'large',
 'datasets',
 'to',
 'improve',
 'decision',
 'making',
 'and',
 'develop',
 'intelligent',
 'systems',
 'that',
 'solve',
 'complex',
 'problems',
 '.']
```

POS Tagging using NLTK

```
nltk.download('averaged_perceptron_tagger_eng')
pos_nltk = nltk.pos_tag(tokens_nltk)
pos_nltk
```

```
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger_eng.zip.
[('Artificial', 'JJ'),
 ('Intelligence', 'NNP'),
 ('has', 'VBZ'),
 ('transformed', 'VBN'),
 ('modern', 'JJ'),
 ('research', 'NN'),
 ('.', '.'),
 ('Researchers', 'NNP'),
 ('analyze', 'VBP'),
 ('large', 'JJ'),
 ('datasets', 'NNS'),
 ('to', 'TO'),
 ('improve', 'VB'),
 ('decision', 'NN'),
 ('making', 'NN'),
 ('and', 'CC'),
 ('develop', 'VB'),
 ('intelligent', 'JJ'),
 ('systems', 'NNS'),
 ('that', 'WDT'),
 ('solve', 'VBP'),
 ('complex', 'JJ'),
 ('problems', 'NNS'),
 ('.', '.')]
```

POS Tagging using spaCy

```
doc = nlp(text)
[(token.text, token.pos_) for token in doc]
```

```
[('Artificial', 'PROPN'),
 ('Intelligence', 'PROPN'),
 ('has', 'AUX'),
 ('transformed', 'VERB'),
 ('modern', 'ADJ'),
 ('research', 'NOUN'),
 ('.', 'PUNCT'),
 ('Researchers', 'NOUN'),
 ('analyze', 'VERB'),
 ('large', 'ADJ'),
 ('datasets', 'NOUN'),
 ('to', 'PART'),
 ('improve', 'VERB'),
 ('decision', 'NOUN'),
 ('making', 'NOUN'),
 ('and', 'CCONJ'),
 ('develop', 'VERB'),
 ('intelligent', 'ADJ'),
 ('systems', 'NOUN'),
 ('that', 'PRON'),
 ('solve', 'VERB'),
 ('complex', 'ADJ'),
 ('problems', 'NOUN'),
 ('.', 'PUNCT')]
```

Extract Nouns and Verbs (spaCy)

```
nouns = [token.text for token in doc if token.pos_ == 'NOUN']
verbs = [token.text for token in doc if token.pos_ == 'VERB']
nouns, verbs
```

```
(['research',
  'Researchers',
```

```
 'datasets',
 'decision',
 'making',
 'systems',
 'problems'],
['transformed', 'analyze', 'improve', 'develop', 'solve'])
```

Frequency Analysis

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

```
(Counter({'research': 1,
          'Researchers': 1,
          'datasets': 1,
          'decision': 1,
          'making': 1,
          'systems': 1,
          'problems': 1}),
 Counter({'transformed': 1,
          'analyze': 1,
          'improve': 1,
          'develop': 1,
          'solve': 1}))
```

Comparison of Tag Sets

- NLTK uses Penn Treebank tags (NN, VB, JJ)
- spaCy uses universal tags (NOUN, VERB, ADJ)

Discussion

This analysis shows how academic texts rely heavily on nouns to express concepts and verbs to build arguments. spaCy provides cleaner universal tags, while NLTK provides detailed syntactic tags useful for linguistic analysis.