Stop Words in NLP + Stemming + Lemmatization

What are Stop Words?

Stop words are common words in a language such as 'the', 'is', 'in', 'and', etc., which are usually removed during text preprocessing because they add little semantic value.

Why Remove Stop Words?

- To reduce noise in text data.

- To decrease dimensionality of feature space.

- To improve computational efficiency.

1. Using NLTK Stop Words

NLTK provides a corpus of stop words. Key functions/methods:

- stopwords.words(): Get the list of stop words for a language.

- word_tokenize(): Tokenize a sentence.

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

text = "This is an example showing off stop word filtration."

stop_words = set(stopwords.words('english'))
words = word_tokenize(text)

filtered = [w for w in words if w.lower() not in stop_words]

print(filtered)  # ['example', 'showing', 'stop', 'word', 'filtration', '.']
```

2. Using spaCy Stop Words

spaCy provides a built-in list of stop words. Key attributes/functions:

- nlp.Defaults.stop_words: Access the stop word list.

- Token.is_stop: Check if a token is a stop word.

- Customize: Add or remove stop words.

```
import spacy
```

```
nlp = spacy.load("en_core_web_sm")
doc = nlp("This is an example showing off stop words.")

for token in doc:
    print(token.text, token.is_stop)

# Example: Add 'showing' to stop words
nlp.Defaults.stop_words.add("showing")
```

3. Using Scikit-learn Stop Words

Scikit-learn provides a built-in list for vectorizers.

- ENGLISH_STOP_WORDS: A frozenset of stop words.

- Can be passed to CountVectorizer or TfidfVectorizer.

```
from sklearn.feature_extraction.text import ENGLISH_STOP_WORDS
from sklearn.feature_extraction.text import CountVectorizer

print(list(ENGLISH_STOP_WORDS)[:10])

vectorizer = CountVectorizer(stop_words='english')
docs = ["This is an example.", "We remove the stop words."]
X = vectorizer.fit_transform(docs)

print(vectorizer.get_feature_names_out())  # ['example' 'remove' 'stop' 'words']
```

4. Custom Stop Words

You can define your own stop word list for special use cases.

Example:

```
custom_stop_words = {'this', 'is', 'an', 'the'}
text = "This is an example of custom stop words."
words = text.lower().split()
filtered = [w for w in words if w not in custom_stop_words]

print(filtered)  # ['example', 'of', 'custom', 'stop', 'words.']
```

5. Stop Words Removal + Stemming

Example program that removes stop words and then applies stemming using NLTK's PorterStemmer:

```
from nltk.corpus import stopwords

from nltk.tokenize import word_tokenize

from nltk.stem import PorterStemmer


text = "This is an example showing how to filter stop words and apply stemming."


stop_words = set(stopwords.words('english'))

words = word_tokenize(text)


# Remove stop words

filtered = [w for w in words if w.lower() not in stop_words]


# Apply stemming

ps = PorterStemmer()

stemmed = [ps.stem(w) for w in filtered]


print("Filtered:", filtered)

print("Stemmed:", stemmed)
```

6. Stop Words Removal + Lemmatization

Example program that removes stop words and then applies lemmatization using NLTK's WordNetLemmatizer:

```
from nltk.corpus import stopwords, wordnet

from nltk.tokenize import word_tokenize

from nltk.stem import WordNetLemmatizer


text = "The children are playing happily while the dogs are barking."


stop_words = set(stopwords.words('english'))

words = word_tokenize(text)
```

```
# Remove stop words
filtered = [w for w in words if w.lower() not in stop_words]

# Apply lemmatization
lemmatizer = WordNetLemmatizer()
lemmatized = [lemmatizer.lemmatize(w, pos='v') for w in filtered]

print("Filtered:", filtered)
print("Lemmatized:", lemmatized)
```