

Perform Named Entity Recognition (NER) on a given text.

```
In [1]: import spacy
nlp = spacy.load("en_core_web_sm")
```

```
In [2]: text = "Apple Inc. is looking at buying U.K. startup for $1 billion. Tim Cook is the CEO."
doc = nlp(text)
```

```
In [3]: print("Named Entities:")
for ent in doc.ents:
    print(f"{ent.text} ({ent.label_})")
```

Named Entities:
Apple Inc. (ORG)
U.K. (GPE)
\$1 billion (MONEY)
Tim Cook (PERSON)

```
In [4]: from spacy import displacy

displacy.render(doc, style="ent")
```

Apple Inc. **ORG** is looking at buying U.K. **GPE** startup for \$1 billion **MONEY** . Tim Cook **PERSON** is the CEO.

```
In [5]: # Tokenization and BoW
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import CountVectorizer
import nltk
nltk.download('punkt')

tokens = word_tokenize(text)
print("Tokens:", tokens)

vectorizer = CountVectorizer()
X = vectorizer.fit_transform([text])
print("Vocabulary:", vectorizer.get_feature_names_out())
print("BoW Matrix:", X.toarray())
```

Tokens: ['Apple', 'Inc.', 'is', 'looking', 'at', 'buying', 'U.K.', 'startup', 'for', '\$', '1', 'billion', '.', 'Tim', 'Cook', 'is', 'the', 'CEO', '.']
Vocabulary: ['apple' 'at' 'billion' 'buying' 'ceo' 'cook' 'for' 'inc' 'is' 'looking' 'startup' 'the' 'tim']
BoW Matrix: [[1 1 1 1 1 1 1 2 1 1 1 1]]

```
[nltk_data] Downloading package punkt to
[nltk_data]   C:\Users\ASUS\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```