

<https://www.nltk.org/index.html>

<https://www.nltk.org/book/>

```
!pip install nltk
```

```
Requirement already satisfied: nltk in /opt/python/envs/default/lib/python3.11/site-packages/nltk-3.8.1-py3.11.egg
Requirement already satisfied: click in /opt/python/envs/default/lib/python3.11/site-packages/click-8.1.7-py3.11.egg
Requirement already satisfied: joblib in /opt/python/envs/default/lib/python3.11/site-packages/joblib-1.3.2-py3.11.egg
Requirement already satisfied: regex<=2021.8.3 in /opt/python/envs/default/lib/python3.11/site-packages/regex-2021.8.3-py3.11.egg
Requirement already satisfied: tqdm in /opt/python/envs/default/lib/python3.11/site-packages/tqdm-4.64.1-py3.11.egg
```

```
[notice] A new release of pip is available: 23.1.2 -> 23.2.1
```

```
[notice] To update, run: pip install --upgrade pip
```

```
import nltk
```

```
nltk.download("punkt")
nltk.download("stopwords")
```

```
text="NLTK is a leading platform for building Python programs \
to work with human language data. It provides easy-to-use interfaces \
to over 50 corpora and lexical resources such as WordNet, along with a \
suite of text processing libraries for classification, tokenization, stemm\
tagging, parsing, and semantic reasoning, wrappers for industrial-strength\
Thanks to a hands-on guide introducing programming fundamentals alongside\
computational linguistics, plus comprehensive API documentation, NLTK is s\
linguists, engineers, students, educators, researchers, and industry users\
is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free
```

```
words = nltk.word_tokenize(text)
print(words)
```

```
['NLTK', 'is', 'a', 'leading', 'platform', 'for', 'building', 'Python',
```

```
[nltk_data] Downloading package punkt to /home/datalore/nltk_data...
```

```
[nltk_data] Package punkt is already up-to-date!
```

```
[nltk_data] Downloading package stopwords to
```

```
[nltk_data] /home/datalore/nltk_data...
```

```
[nltk_data] Package stopwords is already up-to-date!
```

```
!pip install wordcloud
```

```
Requirement already satisfied: wordcloud in /opt/python/envs/default/lib/python3.11/site-packages/wordcloud-1.9.1-py3.11.egg
Requirement already satisfied: numpy>=1.6.1 in /opt/python/envs/default/lib/python3.11/site-packages/numpy-1.24.3-py3.11.egg
Requirement already satisfied: pillow in /opt/python/envs/default/lib/python3.11/site-packages/pillow-9.5.0-py3.11.egg
Requirement already satisfied: matplotlib in /opt/python/envs/default/lib/python3.11/site-packages/matplotlib-3.7.1-py3.11.egg
Requirement already satisfied: contourpy>=1.0.1 in /opt/python/envs/default/lib/python3.11/site-packages/contourpy-1.0.7-py3.11.egg
```

```
[notice] A new release of pip is available: 23.1.2 -> 23.2.1
[notice] To update, run: pip install --upgrade pip
```

[Download](#)



```
('I', 'love')
('love', 'natural')
('natural', 'language')
('language', 'processing')
```

```
skip_grams=list(skipgrams(tokens,2,3))
for sg in skip_grams:
    print(sg)
```

```
('I', 'love')
('I', 'natural')
('I', 'language')
('I', 'processing')
('love', 'natural')
('love', 'language')
('love', 'processing')
('natural', 'language')
('natural', 'processing')
('language', 'processing')
```

Stemming Stemming is a technique used to reduce an inflected word down to its word stem. For example, the words “programming,” “programmer,” and “programs” can all be reduced down to the common word stem “program.” In other words, “program” can be used as a synonym for the prior three inflection words.

```
from nltk.stem import PorterStemmer
nltk.download("punkt")
ps=PorterStemmer()
example_wordlists=["programming","programmer","programs","programmed"]
for word in example_wordlists:
    print(word, ps.stem(word))
```

```
programming program
programmer programm
programs program
programmed program
```

```
[nltk_data] Downloading package punkt to /home/datalore/nltk_data...
[nltk_data] Package punkt is already up-to-date!
```

Lemmatization is another technique used to reduce inflected words to their root word. It describes the algorithmic process of identifying an inflected word’s “lemma” (dictionary form) based on its intended meaning.

```
from nltk.stem import WordNetLemmatizer
nltk.download("wordnet")
wnl = WordNetLemmatizer()
example_wordlists2=["programming","programmer","programs","programmed"]
for word in example_wordlists:
    print(word, wnl.lemmatize(word,pos="v"))
```

```
programming program
programmer programmer
```

programs program
programmed program

[nltk_data] Downloading package wordnet to /home/datalore/nltk_data...
[nltk_data] Package wordnet is already up-to-date!