

NLTK vs spaCy

NLTK vs spaCy: Feature Comparison

Feature / Aspect	NLTK (Natural Language Toolkit)	spaCy
Philosophy	Research & teaching tool	Industrial-strength NLP for production
Ease of Use	Beginner-friendly, very flexible	High-level, fast, but less customizable
Speed	Slower (Python-based, older architecture)	Very fast (Cython-optimized)
Accuracy	Lower, older models	High, with modern deep learning models
Installation Size	Lightweight core; corpora/models downloaded separately	Comes with medium-size models (~50 MB+)
Pretrained Models	Basic ones (POS, NER, WordNet, etc.)	Strong multilingual models, transformer support
Deep Learning Support	Minimal (external tools needed)	Built-in transformers & pipelines via <code>spacy-transformers</code>
Customization	Very flexible and modular	Less modular, but pipelines are customizable
Best Use Cases	Education, prototyping, language experiments	Production NLP, performance-critical apps
Tokenization	Rule-based	Statistical + rule-based, very robust
POS Tagging	Averaged Perceptron, simple	Trained on large corpora, more accurate
Named Entity Recognition (NER)	Basic	Advanced, with entity linking support
Dependency Parsing	Available but basic	Accurate and production-ready

Feature / Aspect	NLTK (Natural Language Toolkit)	spaCy
Corpora Support	Large collection (e.g., Gutenberg, movie reviews, WordNet)	Minimal built-in corpora
Integration	Plays well with NLTK corpora, older tools	Integrates with Hugging Face, PyTorch, etc.

Capabilities

NLP Task	NLTK	spaCy
Tokenization	Yes	Yes (much faster and more robust)
POS Tagging	Yes (multiple taggers)	Yes (state-of-the-art, trained models)
Named Entity Recognition (NER)	Yes (basic)	Yes (advanced and more accurate)
Dependency Parsing	Yes (but slower and less accurate)	Yes (very efficient and accurate)
Lemmatization	Yes (WordNetLemmatizer)	Yes (built-in and faster)
Sentiment Analysis	Basic (e.g., VADER)	No built-in (use external models)
Word Embeddings	No built-in	Yes (supports word vectors like GloVe)
Custom Pipelines	Limited	Yes (customizable processing pipelines)

When to Use What?

If you want to...	Use
Learn NLP basics, explore corpora, research	NLTK
Build fast, modern NLP pipelines for production	spaCy
Work with WordNet or classic NLP datasets	NLTK

If you want to...	Use
Do high-performance NER, POS tagging, parsing	spaCy

Ecosystem & Integration

Feature	NLTK	spaCy
Pre-trained Models	Minimal	Rich, language-specific models
Integration with ML Tools	Manual	Works well with <code>scikit-learn</code> , <code>TensorFlow</code> , <code>PyTorch</code>
Corpus Support	Huge collection of corpora	Minimal, but supports external sources

Code Comparison Example

Tokenization:

NLTK:

```
from nltk.tokenize import word_tokenize
word_tokenize("I love NLP!")
```


```
['I', 'love', 'NLP', '!']
```

spaCy:

```
import spacy
nlp = spacy.load("en_core_web_sm")
doc = nlp("I love NLP!")
[token.text for token in doc]
```

```
['I', 'love', 'NLP', '!']
```

Summary

| For Learning/Academia |  **NLTK** |

| For Real-World Apps |  **spaCy** |