spaCy

Industrial-Strength Natural Language Processing

IN PYTHON

Fastest in the world

spaCy excels at large-scale information extraction tasks. It's written from the ground up in carefully memory-managed Cython. Independent research has confirmed that spaCy is the fastest in the world. If your application needs to process entire web dumps, spaCy is the library you want to be using.

FACTS & FIGURES

Get things done

spaCy is designed to help you do real work — to build real products, or gather real insights. The library respects your time, and tries to avoid wasting it. It's easy to install, and its API is simple and productive. We like to think of spaCy as the Ruby on Rails of Natural Language Processing.

GET STARTED

Deep learning

spaCy is the best way to prepare text for deep learning. It interoperates seamlessly with TensorFlow, PyTorch, scikit-learn, Gensim and the rest of Python's awesome AI ecosystem. With spaCy, you can easily construct linguistically sophisticated statistical models for a variety of NLP problems.

READ MORE

spaCy　　　　　　　　　　　　　　**USAGE**　　**MODELS**　　**API**　　

LIGHTNING_TOUR.PY

```python
# Install: pip install spacy && python -m spacy download en
import spacy

# Load English tokenizer, tagger, parser, NER and word vectors
nlp = spacy.load('en')

# Process whole documents
text = open('customer_feedback_627.txt').read()
doc = nlp(text)

# Find named entities, phrases and concepts
for entity in doc.ents:
    print(entity.text, entity.label_)

# Determine semantic similarities
doc1 = nlp(u'the fries were gross')
doc2 = nlp(u'worst fries ever')
doc1.similarity(doc2)

# Hook in your own deep learning models
nlp.add_pipe(load_my_model(), before='parser')
```

MORE EXAMPLES

Features

- Non-destructive **tokenization**

spaCy

- **13 statistical models** for 8 languages
- Pre-trained **word vectors**
- Easy **deep learning** integration
- Part-of-speech tagging
- Labelled dependency parsing
- Syntax-driven sentence segmentation
- Built in **visualizers** for syntax and NER
- Convenient string-to-hash mapping
- Export to numpy data arrays
- Efficient binary serialization
- Easy **model packaging** and deployment
- State-of-the-art speed
- Robust, rigorously evaluated accuracy

NEW IN V2.0

spaCy

Convolutional neural network models

spaCy v2.0 features new neural models for **tagging**, **parsing** and **entity recognition**. The models have been designed and implemented from scratch specifically for spaCy, to give you an unmatched balance of speed, size and accuracy. A novel bloom embedding strategy with subword features is used to support huge vocabularies in tiny tables. Convolutional layers with

residual connections, layer normalization and maxout non-linearity are used, giving much better efficiency than the standard BiLSTM solution. Finally, the parser and NER use an imitation learning objective to deliver accuracy in-line with the latest research systems, even when evaluated from raw text. With these innovations, spaCy v2.0's models are **10× smaller**, **20% more accurate**, and **even cheaper to run** than the previous generation.

DOWNLOAD MODELS

SPACY IS TRUSTED BY

spaCy

USAGE   MODELS   API

FEATURED ON

recode   The Washington Post   B B C   Microsoft

VentureBeat   ThoughtWorks®

FROM THE MAKERS OF SPACY

Prodigy: Radically efficient machine teaching

spaCy

Prodigy is an **annotation tool** so efficient that data scientists can do the annotation themselves, enabling a new level of rapid iteration. Whether you're working on entity recognition, intent detection or image classification, Prodigy can help you **train and evaluate** your models faster. Stream in your own examples or real-world data from live APIs, update your model in real-time and chain models together to build more complex systems.

TRY IT OUT

Benchmarks

In 2015, independent researchers from Emory University and Yahoo! Labs showed that spaCy offered the **fastest syntactic parser in the world** and that its accuracy was **within 1% of the best** available (Choi et al., 2015). spaCy v2.0, released in 2017, is more accurate than any of the systems Choi et al. evaluated.

SEE DETAILS