

SPACY

FERNANDO ARTURO AREVALO PEREZ
MARIA DEL CARMEN HERRERA
JIMENEZ



¿Qué es spaCy?



Biblioteca de código abierto para Procesamiento de Lenguaje Natural (NLP)



Desarrollada en Python y Cython



Enfocada en producción y aplicaciones reales



Una de las bibliotecas NLP más rápidas disponibles

Características Principales

Tokenización avanzada

Reconocimiento de entidades nombradas
(NER)

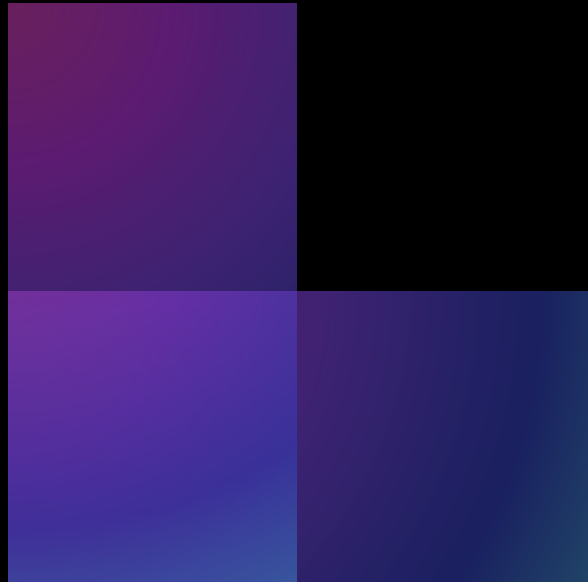
Etiquetado gramatical (POS Tagging)

Análisis de dependencias

Word Vectors

Compatibilidad con más de 64 idiomas

Fácil integración con Deep Learning



Casos de Uso Reales

**Análisis de
Sentimientos**

Redes sociales

Opiniones de
clientes

Reviews de
productos

**Extracción de
Información**

Procesamiento
de CVs

Análisis de
documentos
legales

Extracción de
datos de
facturas

**Clasificación
de Textos**

Categorización
de emails

Clasificación
de
documentos

Filtrado de
spam

Ventajas de spaCy



✓ Rápido y eficiente



✓ Fácil de aprender



✓ Bien documentado



✓ Modelos pre-entrenados



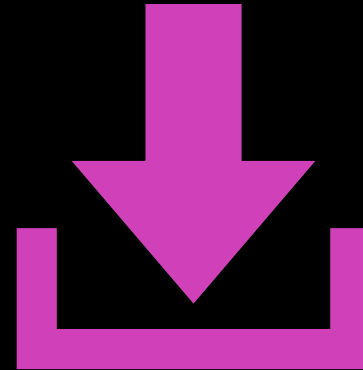
✓ Comunidad activa



✓ Integración con otras herramientas

1. Instalación

```
▶ # Instalar spaCy  
!pip install spacy  
  
# Descargar modelo en español  
!python -m spacy download es_core_news_sm  
  
# Descargar modelo en inglés  
!python -m spacy download en_core_web_sm  
[4] ✓ 38.5s
```



```
▶ import spacy  
  
# Cargar el modelo en español  
nlp = spacy.load("es_core_news_sm")  
  
# Procesar un texto  
texto = "Apple está buscando comprar una startup del Reino Unido por mil millones de dólares."  
doc = nlp(texto)  
[5] ✓ 1.7s
```

Funcionalidades Básicas

1. Tokenización
2. Análisis Gramatical (POS)
3. Reconocimiento de Entidades (NER)

1. Tokenización

```
▶ # Mostrar parte del discurso
for token in doc:
    print(f"{token.text}: {token.pos_}")
```

[7] ✓ 0.0s

... Apple: PROPN
está: AUX
buscando: VERB
comprar: VERB
una: DET
startup: NOUN
del: ADP
Reino: PROPN
Unido: PROPN
por: ADP
mil: NUM
millones: NOUN
de: ADP
dólares: NOUN
.: PUNCT

2. Análisis Gramatical (POS)

```
▶ # Dividir texto en tokens
for token in doc:
    print(token.text)
```

[6] ✓ 0.0s

... Apple
está
buscando
comprar
una
startup
del
Reino
Unido
por
mil
millones
de
dólares
.

3. Reconocimiento de Entidades (NER)

¿Qué detecta?

- Nombres de personas
- Organizaciones
- Ubicaciones
- Fechas
- Cantidades
- Y más...

```
▶ # Detectar entidades
for ent in doc.ents:
    print(f"Texto: {ent.text}, Tipo: {ent.label_}")
```

[8] ✓ 0.0s

... Texto: Apple, Tipo: ORG
Texto: Reino Unido, Tipo: LOC

Ejemplo práctico NER

- Entidades nombradas
- Análisis gramatical
- Dependencias sintácticas

```
import spacy

# Cargar el modelo en español
nlp = spacy.load("es_core_news_sm")

• texto = """
Microsoft anunció ayer en Nueva York una inversión
de 1000 millones de dólares en tecnología AI.
"""

doc = nlp(texto)
```

```
# 1. Imprimir entidades nombradas
print("📋 ENTIDADES ENCONTRADAS:")
for ent in doc.ents:
    print(f"- Texto: {ent.text}")
    print(f"  Tipo: {ent.label_}")
    print(f"  Explicación: {spacy.explain(ent.label_)}\n")

# 2. Imprimir análisis gramatical
print("\n📋 ANÁLISIS GRAMATICAL:")
for token in doc:
    print(f"- Palabra: {token.text}")
    print(f"  Tipo: {token.pos_}")
    print(f"  Explicación: {spacy.explain(token.pos_)}\n")

# 3. Imprimir dependencias sintácticas
print("\n📋 DEPENDENCIAS SINTÁCTICAS:")
for token in doc:
    print(f"- Palabra: {token.text}")
    print(f"  Dependencia: {token.dep_}")
    print(f"  Explicación: {spacy.explain(token.dep_)}\n")
```

✓ 1.1s

ENTIDADES ENCONTRADAS:

- Texto: Microsoft
Tipo: ORG
Explicación: Companies, agencies, institutions, etc.
- Texto: Nueva York
Tipo: LOC
Explicación: Non-GPE locations, mountain ranges, bodies of water
- Texto: AI
Tipo: ORG
Explicación: Companies, agencies, institutions, etc.

ANÁLISIS GRAMATICAL:

- Palabra:

Tipo: SPACE
Explicación: space
- Palabra: Microsoft
Tipo: PROPN
Explicación: proper noun
- Palabra: anunció
Tipo: VERB
Explicación: verb
- Palabra: ayer
Tipo: ADV
Explicación: adverb

DEPENDENCIAS SINTÁCTICAS:

- Palabra:

Dependencia: dep
Explicación: unclassified dependent
- Palabra: Microsoft
Dependencia: nsubj
Explicación: nominal subject
- Palabra: anunció
Dependencia: ROOT
Explicación: root
- Palabra: ayer
Dependencia: advmod
Explicación: adverbial modifier
- Palabra: en
Dependencia: case
Explicación: case marking

Práctica de funciones de spacy

0. Primeros pasos con spaCy
1. Tokenización
2. Lematización
3. Reconocimiento de Entidades (NER)
4. Patrones (Matching) - Patrones (Matching) 2da parte
5. Etiquetado de Partes del Discurso (POS Tagging)
6. Análisis de Dependencias Sintácticas
7. Vectorización y Similitud de Textos
8. Clasificación de Texto – Hugging Face Transformers – Análisis de Sentimientos
9. Integración con Otros Frameworks y Modelos
10. Extensibilidad y Personalización
11. Soporte Multilingüe
12. Documentación y Comunidad Activa