Introduzione a NLTK e SpaCy

CORSO DI NATURAL LANGUAGE PROCESSING (ELABORAZIONE DEL LINGUAGGIO NATURALE)

a.a. 2025/2026

Prof. Roberto Pirrone

Cos'è NLTK (Natural Language ToolKit)?

- Piattaforma leader per la costruzione di programmi Python per l'analisi del linguaggio umano.

- Nato in ambito accademico e di ricerca, è un toolkit estremamente completo e modulare.

- Il suo scopo principale è supportare l'insegnamento e la ricerca nel campo del Natural Language Processing (NLP) e della linguistica computazionale.

Funzionalità Principali

- *Accesso a Corpora e Risorse Lessicali*: Include decine di corpus e lessici pronti all'uso (es. WordNet).
- *Tokenizzazione*: Suddivisione del testo in parole o frasi.
- *Stemming & Lemmatizzazione*: Riduzione delle parole alla loro radice (es. "camminando" -> "cammin").

Funzionalità Principali

- *Part-of-Speech Tagging (POS)*: Etichettatura grammaticale.
- *Parsing (Analisi Sintattica)*: Creazione di alberi sintattici per analizzare la struttura delle frasi.
- Classificazione del Testo e algoritmi di Machine Learning classici.

Architettura

- A differenza di pipeline integrate come SpaCy, NLTK è una collezione di moduli indipendenti.
- L'utente sceglie e combina i tool di cui ha bisogno per costruire la propria applicazione.
- Garantisce flessibilità e trasparenza nell'analisi della pipeline

Risorse utili

- Sito Ufficiale: www.nltk.org
- Il Libro di NLTK (gratuito online): www.nltk.org/book/ (una risorsa fondamentale per imparare)
- GitHub Repository: github.com/nltk/nltk

SpaCy

- *Cos'è SpaCy?* SpaCy è una libreria open-source per l'elaborazione avanzata del linguaggio naturale (NLP) in Python. Progettata per essere efficiente e pronta per la produzione.

- *Obiettivo*: Offrire gli strumenti più veloci e accurati per il NLP moderno, focalizzandosi su prestazioni, facilità d'uso e supporto per modelli pre-addestrati.

- *Differenza chiave*: Non è un toolkit di ricerca (come NLTK), ma una libreria orientata all'applicazione, con API pulite e architetture ottimizzate.

Funzionalità Principali di SpaCy

- *Tokenizzazione*: Suddivisione del testo in unità significative (parole, punteggiatura).
    - *Differenza da NLTK*: Tokenizer non-distruttivo, specifico per lingua e altamente accurato, capace di gestire contrazioni, abbreviazioni e valute.
- *Part-of-Speech Tagging (POS Tagging)*: Assegnazione di una categoria grammaticale (sostantivo, verbo, aggettivo) a ogni token.
- *Named Entity Recognition (NER)*: Identificazione e classificazione di entità denominate (persone, luoghi, organizzazioni, date) nel testo.

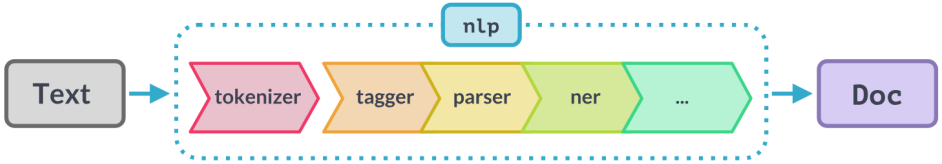
Funzionalità Principali di SpaCy

- *Dependency Parsing*: Analisi delle relazioni grammaticali tra le parole in una frase, costruendo una struttura ad albero.
- *Lemmatizzazione*: Riduzione delle parole alla loro forma base (lemma) (es. "correndo" -> "correre").
- *Vettorizzazione di Parole/Documenti*: Conversione del testo in rappresentazioni numeriche (embedding) per modelli di Machine Learning.
- *Text Classification*: Strumenti per addestrare modelli di classificazione del testo.

Architettura base: La Pipeline NLP

- SpaCy elabora il testo attraverso una "pipeline" di componenti. Ogni componente esegue un'operazione specifica sul documento e arricchisce l'oggetto Doc.
- Componenti Standard:
  - Tokenizer: Il primo componente, crea l'oggetto Doc dai token grezzi.
  - Tagger: Assegna i POS tag.
  - Parser: Esegue il dependency parsing.
  - NER: Rileva le Named Entities.
- La pipeline è modulare; è possibile aggiungere, rimuovere o riordinare i componenti, o creare componenti personalizzati.

Architettura base: La Pipeline NLP



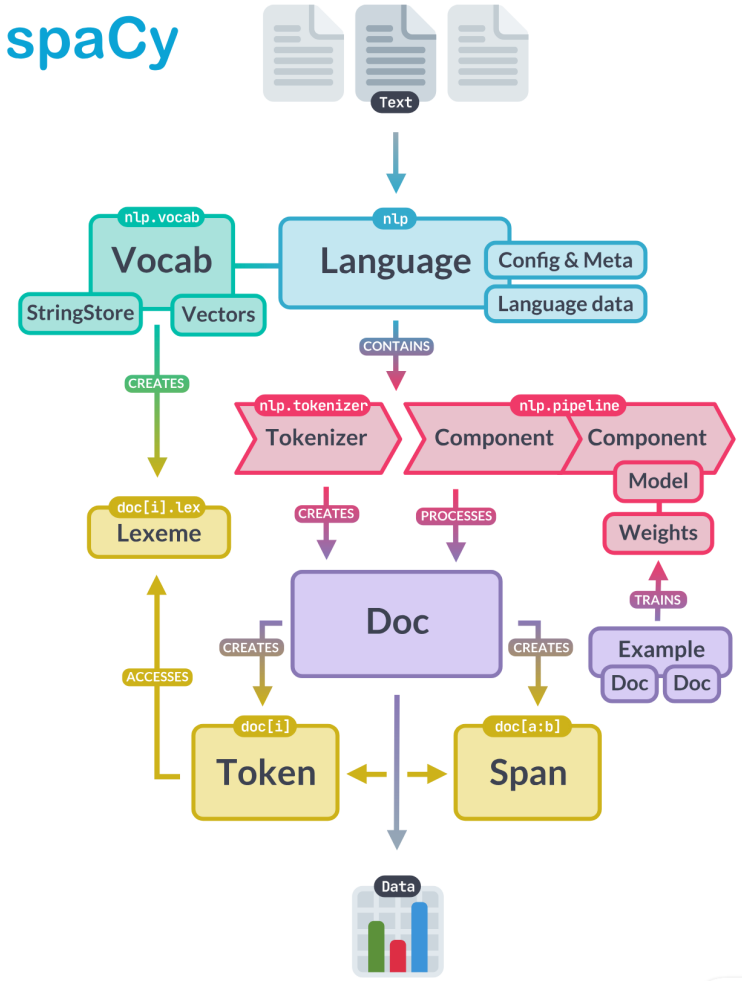
NAME	COMPONENT	CREATES	DESCRIPTION
tokenizer	`Tokenizer`	`Doc`	Segment text into tokens.
PROCESSING PIPELINE			
tagger	`Tagger`	`Token.tag`	Assign part-of-speech tags.
parser	`DependencyParser`	`Token.head`, `Token.dep`, `Doc.sents`, `Doc.noun_chunks`	Assign dependency labels.
ner	`EntityRecognizer`	`Doc.ents`, `Token.ent_iob`, `Token.ent_type`	Detect and label named entities.
lemmatizer	`Lemmatizer`	`Token.lemma`	Assign base forms.
textcat	`TextCategorizer`	`Doc.cats`	Assign document labels.
custom	custom components	`Doc._.xxx`, `Token._.xxx`, `Span._.xxx`	Assign custom attributes, methods or properties.

Università degli Studi di Palermo
dipartimento di ingegneria unipa

CHILAB

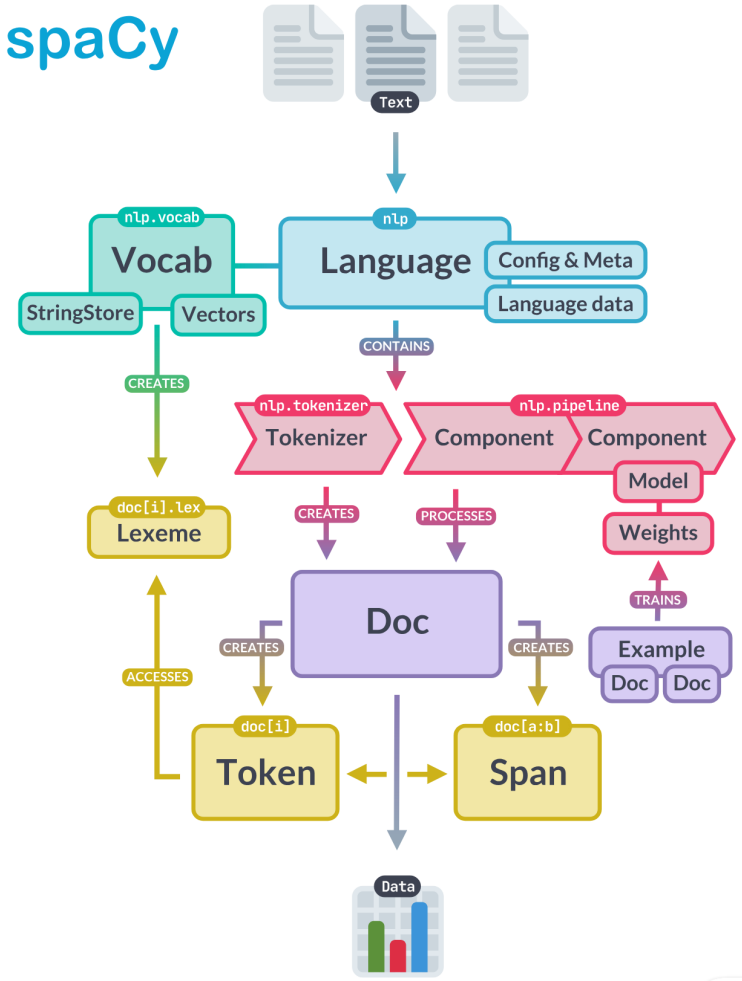
Classi

- *Language Object (`nlp`)*
- È il punto di ingresso principale per l'elaborazione del testo.
- Contiene il vocabolario condiviso (Vocab), i modelli linguistici caricati e la pipeline di elaborazione.
- Quando si scrive nlp("Il testo"), è l'oggetto Language che gestisce l'intero processo.



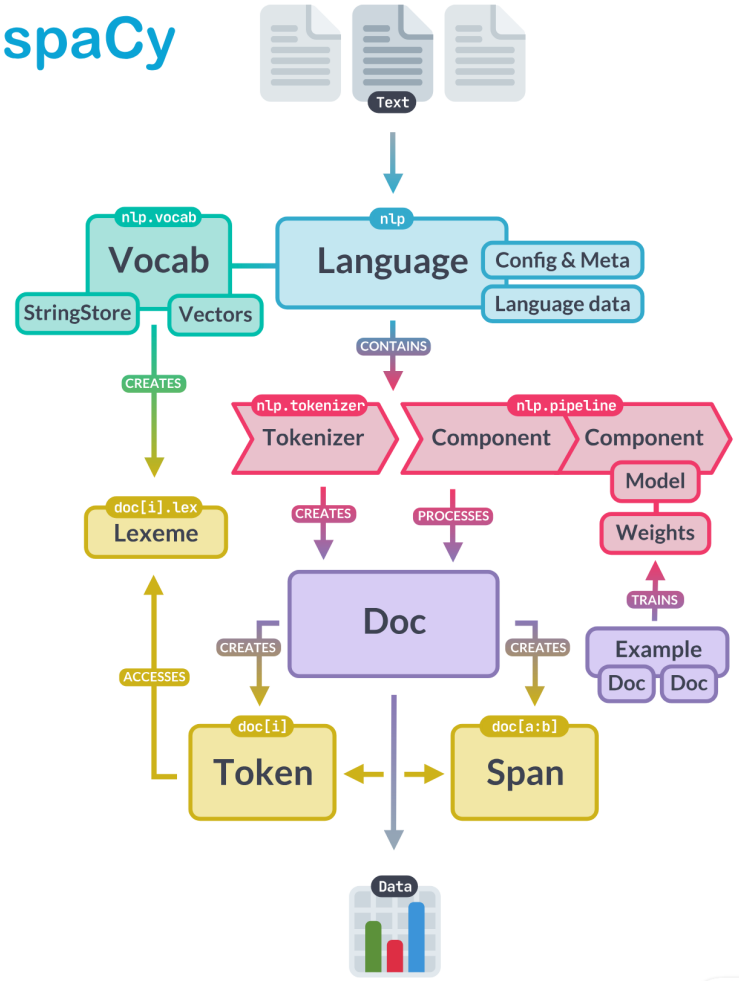
Classi

- *Doc Object*
- La rappresentazione tokenizzata e annotata di un testo.
- È il contenitore principale per tutte le annotazioni linguistiche (token, span, entità, PoS, dipendenze).
- Ogni volta che nlp() elabora un testo, restituisce un oggetto Doc.



Classi

- *Token Object*
- Rappresenta un singolo token all'interno di un Doc.
- Contiene attributi specifici per quel token (es. token.text, token.lemma_, token.pos_).
- *Span Object*
- Rappresenta una sequenza di uno o più Token all'interno di un Doc. Usato spesso per le entità.



Modelli Pre-addestrati e Lingue Supportate

- SpaCy offre modelli pre-addestrati per decine di lingue, inclusi italiano, inglese, tedesco, francese, spagnolo e molti altri.
- Modelli disponibili in diverse dimensioni (small _sm, medium _md, large _lg) per bilanciare velocità, accuratezza e dimensioni del file.
- Possibilità di addestrare i propri modelli per compiti specifici o domini verticali.

Risorse utili

- Sito Ufficiale e Documentazione: https://spacy.io/
- GitHub Repository: github.com/explosion/SpaCy