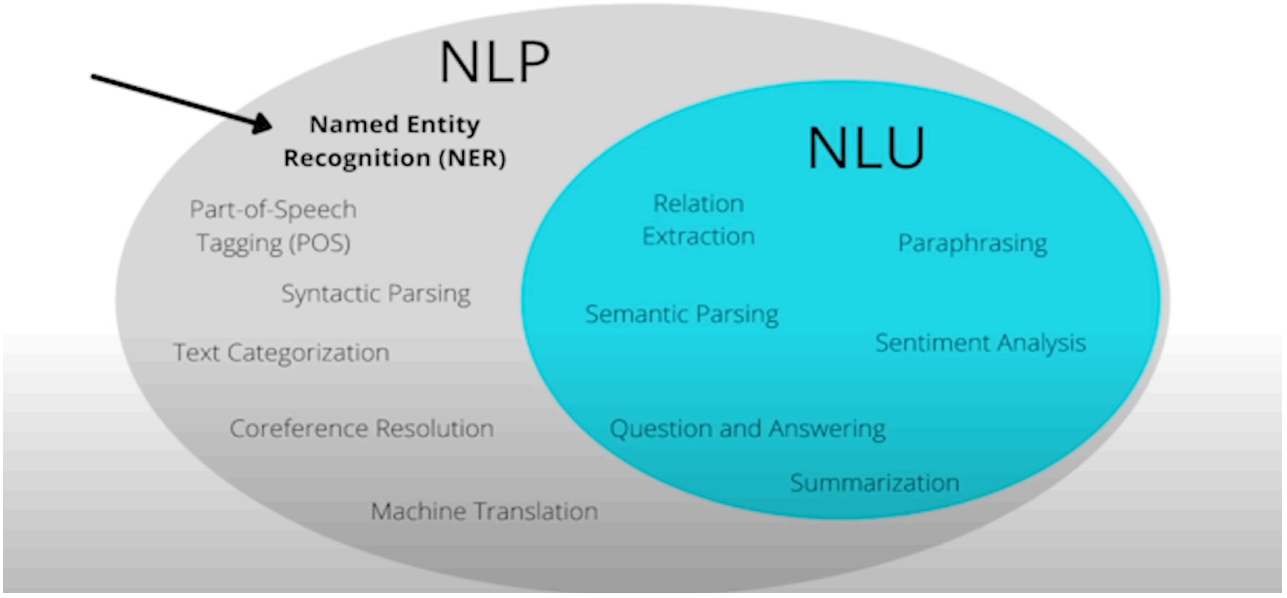
24 - Prática: Processamento de Linguagem Natural (NLP) (III)
Higor Miller Grassi

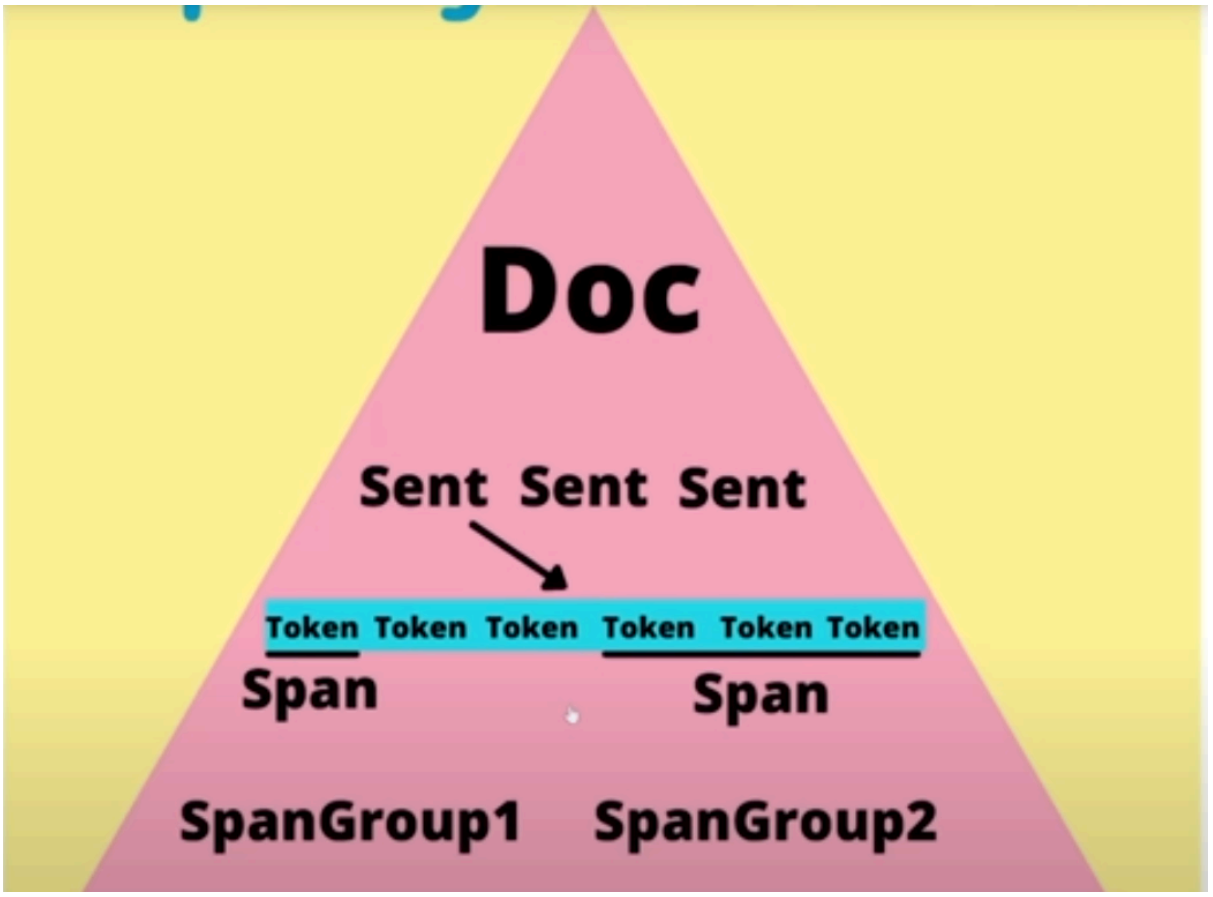
Tokenização: As palavras são formadas por letras, onde as mesmas podem ser representadas por um conjunto de números, assim, tendo o famoso código ASCII. No exemplo dado no vídeo, apenas interpretar as palavras pelas letras pode não ser tão eficaz, já que muitas palavras têm as mesmas letras mas em formações diferentes, então, para diferenciar é numerado as palavras para diferenciar mais facilmente.

NLP (Natural Language Processing): Processo em que a máquina entende, analisa e extrai a linguagem, na maior parte das vezes com um texto bruto. Usando a marcação de classes gramaticais, análise sintática, categorização de texto, resolução de correferências e tradução automática.

LUN (Language Understanding and Normalization): Reduz a complexidade e o tamanho do texto sem perder seu significado essencial. Usado principalmente para realizar extração de relações, análise semântica, perguntas e respostas.



SpaCy: Utilizaremos esta biblioteca pois ela é conhecida por sua rapidez, eficiência e facilidade de uso, tendo modelos pré-treinados para análise sintática, reconhecimento de entidades, categorização de texto e outros recursos avançados, tornando-se ideal para aplicações que exigem processamento rápido e preciso de linguagem natural, escalando muito bem.



A imagem ilustra a estrutura hierárquica de um objeto Doc na biblioteca spaCy, usada para processamento de linguagem natural. No topo da hierarquia está o Doc, que representa um texto completo, esse Doc é composto por sentenças (Sent), que, por sua vez, são formadas por tokens (Token), que representam as menores unidades do texto, como palavras ou pontuações.

Além disso, tokens podem ser agrupados em Span, que representam sequências contínuas de tokens dentro do texto. Vários Span podem ser organizados em SpanGroup, permitindo a categorização de diferentes partes do texto para tarefas específicas, como reconhecimento de entidades nomeadas ou análise sintática.

Named Entity Recognition (NER): Identifica e classifica as entidades mencionadas em textos, como nomes de pessoas, organizações, locais, datas, entre outros.

Word Vectors: Eles capturam o significado das palavras com base no contexto em que aparecem.

Pipelines: É uma sequência de processos aplicados a um texto para transformá-lo em um objeto Doc, que contém tokens, informações linguísticas e anotações.

Entity Ruler: permite adicionar regras personalizadas para identificar e rotular entidades em um texto com base em padrões definidos.

Matcher: Identifica sequências de tokens (palavras ou partes do texto) com base em padrões definidos, diferentemente do entity permite criar padrões complexos para encontrar combinações específicas de tokens, levando em consideração atributos como texto, lema, parte do discurso (POS), dependências sintáticas

Custom Components: são funções ou classes que você pode adicionar ao pipeline de processamento de texto para realizar tarefas específicas, podendo manipular o objeto Doc (que representa o texto processado) e adicionar ou modificar atributos, como entidades, tokens, extensões personalizadas, etc

RegEX_aula: Usados para buscar combinações específicas em textos como palavras números ou sequências de caracteres no spacy o regex pode ser integrado usando ferramentas como o matcher ou o phrase matcher que permitem identificar padrões complexos diretamente no pipeline de processamento de texto.