Tokenization with spaCy - Full Notes

Tokenization is the process of splitting text into smaller units called tokens. In Natural Language Processing (NLP), tokenization can be at the level of words, sentences, or even subwords. spaCy is a popular NLP library in Python that provides efficient and linguistically accurate tokenization.

1. Word Tokenization with spaCy
Word tokenization means splitting text into individual words. spaCy handles punctuation, contractions, and special cases intelligently.

Example:
Input: "I can't believe spaCy handles tokenization so well!"
Output: ['I', 'ca', "n't", 'believe', 'spaCy', 'handles', 'tokenization', 'so', 'well', '!']
Notice that "can't" is split into "ca" and "n't".

2. Sentence Tokenization with spaCy
Sentence tokenization means splitting text into sentences. spaCy uses statistical models and rules to avoid common errors.

Example:
Input: "Dr. Smith lives in New York. He works at Google AI Lab."
Output:
1. Dr. Smith lives in New York.
2. He works at Google AI Lab.
spaCy correctly recognizes that "Dr." is not the end of a sentence.

3. Extra Features in spaCy Tokenization
spaCy tokens carry linguistic attributes such as:
- **is_alpha**: Whether the token is alphabetic.
- **is_stop**: Whether the token is a stopword (e.g., 'the', 'in').
- **lemma_**: The base form of the word (e.g., 'lives' $\rightarrow$ 'live').

Example Output:
Dr. (is_alpha=False, is_stop=False, lemma=Dr.)
Smith (is_alpha=True, is_stop=False, lemma=Smith)
lives (is_alpha=True, is_stop=False, lemma=live)
in (is_alpha=True, is_stop=True, lemma=in)
New (is_alpha=True, is_stop=False, lemma=New)
York (is_alpha=True, is_stop=False, lemma=York)
. (is_alpha=False, is_stop=False, lemma=.)

Summary:
- spaCy Word Tokenization: Splits text into words and punctuation.
- spaCy Sentence Tokenization: Splits text into sentences intelligently.
- spaCy Tokens: Provide linguistic features (POS tags, lemmas, stopwords).

spaCy is highly efficient and accurate for linguistic tokenization, making it ideal for modern NLP applications.