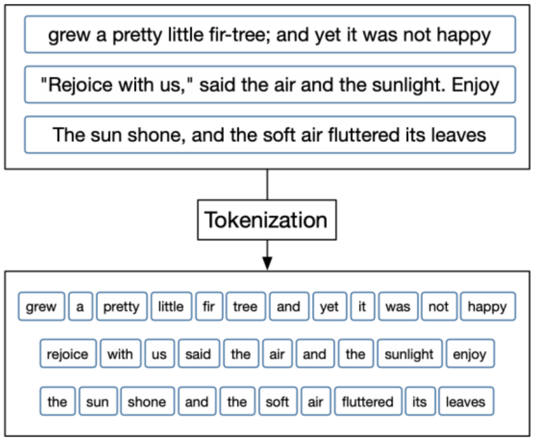
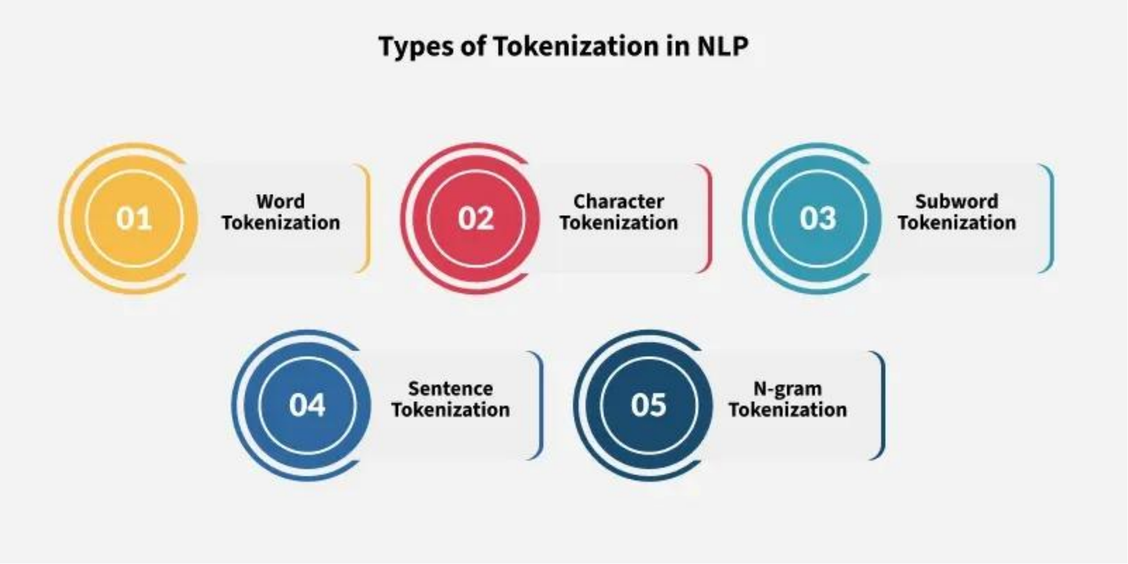
1. What is Tokenization?

Tokenization is the process of breaking raw text into smaller units called tokens. These tokens can represent sentences, words, subwords, or characters. Tokenization transforms unstructured text into a structured format that machines can understand and analyze.


grew a pretty little fir-tree; and yet it was not happy
"Rejoice with us," said the air and the sunlight. Enjoy
The sun shone, and the soft air fluttered its leaves
Tokenization
grew | a | pretty | little | fir | tree | and | yet | it | was | not | happy
rejoice | with | us | said | the | air | and | the | sunlight | enjoy
the | sun | shone | and | the | soft | air | fluttered | its | leaves

2. Why Tokenization is Important

Tokenization plays a critical role in NLP pipelines. It converts raw text into manageable units, helps in building vocabularies, enables numerical representation of text, and significantly impacts the performance and accuracy of NLP models.



Types of Tokenization

1. Sentence Tokenization

Sentence tokenization divides a block of text into individual sentences. It is commonly used in text summarization, document analysis, and dialogue systems. Challenges include handling abbreviations, decimal numbers, and punctuation marks.

2. Word Tokenization

Word tokenization splits sentences into words. It is widely used in classical NLP tasks such as sentiment analysis and topic modeling. However, it faces challenges with contractions, compound words, and multilingual text.

3. Subword Tokenization

Subword tokenization breaks words into smaller meaningful units. It effectively handles out-of-vocabulary words and reduces vocabulary size. Popular subword techniques include Byte Pair Encoding (BPE), WordPiece, and Unigram Language Models.

4. Character Tokenization

Character tokenization splits text into individual characters. While it eliminates unknown words, it increases sequence length and computational cost. It is useful for languages with rich morphology and spelling variations.

5. N-gram Tokenization

N-gram tokenization splits words into fixed-sized chunks (size = n) of data.

Input before tokenization: ["Machine learning is powerful"]

Output when tokenized by bigrams: [('Machine', 'learning'), ('learning', 'is'), ('is', 'powerful')]

Tokenization in Deep Learning

Modern deep learning models such as Transformers rely heavily on subword tokenization. Tokenizers are trained along with model vocabularies to balance efficiency and semantic understanding.

Challenges in Tokenization

Tokenization faces several challenges including ambiguity, multilingual support, emojis, special characters, and context-sensitive meanings. Choosing the correct tokenization strategy is crucial for optimal model performance.

Limitations of Tokenization

- Unable to capture the meaning of the sentence hence, results in ambiguity.
- Chinese, Japanese, Arabic, lack distinct spaces between words. Hence, absence of clear boundaries that complicates the process of tokenization.
- Tough to decide how to tokenize text that may include more than one word, for example email address, URLs and special symbols