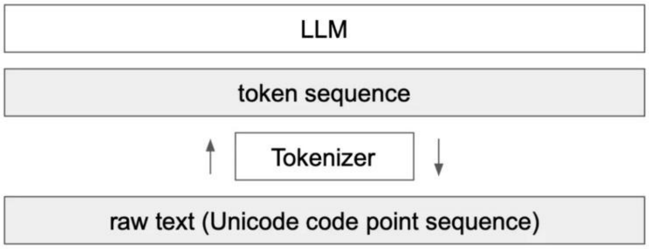
Tokenizer

Tokenizer converts textual data into representation that our network can understand, manipulate and extract information from. It works by encoding and decoding the textual data into token sequences.

Note: A tokenizer is a complete separate and independent module from LLM. Training a tokenizer is different from training LLM. It encodes the Textual data into token sequences that LLM can train on, and also decodes the output token sequences generated by LLM into text.



The tokenizer used by OpenAI models uses tokenizing algorithm called Byte-Pair Encoding.

Byte Pair Encoding (BPE): A very simple tokenization algorithm

1. Start with characters as tokens.
2. Count all adjacent token pairs.
3. Merge the most frequent pair into a new token.
4. Repeat until desired vocab size.
5. Use the learned merges to tokenize new text.

Note: A regex is used to make sure some characters never end up merging together. Like letters with punctuation etc.

TikToken: Its the byte-level tokenizer by OpenAI used to train GPT models. It uses the BPE algorithm. It only provides Inference and not training. It uses pretrained vocab for encoding and decoding based on openai merge rules. Its only a runtime tokenizer.

SentencePiece: It's the tokenizer used to train llama models. Its available as open source for both training and inference. It can utilize BPE as well as Unigram language modelling (probabilistic subword model for Unicode). It differs from tiktoken because it applies tokenization on Unicode characters (UTF-8). It doesn't handle non-UTF directly and has to rely on training fallback to use BPE merges.