Tokenization (Hands on in Jupyter notebook)

Basic block for building LLM

① How do we prepare input text for training LLMs?

1. Splitting text into individual word and subword tokens
2. Convert tokens into token IDs
3. Encode token IDs into vector representation

② Let us look at step 1: Tokenizing text

Output Text

Post Processing steps

GPT

□□□ □□□ □□□ □□□ Token Embeddings (step 3)

4013 | 201 | 302 | 1134 — Token IDs (step 2)

This | is | an | example — Tokenized text (step 1)

This is an example — Input Text

(a) Dataset used: "The verdict" by Edith Wharton

(b) Download and Load in Python

Complete training dataset → Tokenized text → vocabulary (sorted alphabetically)

"The quick brown fox jumps over the lazy dog"

The quick brown fox

Each unique token is mapped to a unique integer called token ID

brown - 0
dog - 1
fox - 2
jumps - 8
lazy - 4
over - 5
quick - 6
the - 7

token IDs

unique tokens

(c) Implement the tokenizer class in Python

Sample text! → Tokenized text → Token IDs

The fox chased the dog → The fox chased the dog → 3 2 0 3 1

Encode

ENCODE

tokenizer encode(text)

DECODE

Token IDs → Tokenized text → sample text

3 2 0 3 1 → The fox chased the dog → The fox chased the dog

tokenizer decode (IDs)

Special Context Tokens

- used to handle unknown words
- Modify the vocab and tokenizer we implemented in previous section (2.2)

(8.) Adding special text tokens

"The fox chased the dog quickly"

→ [The] [fox] [chased] [the] [dog]

Existing vocab

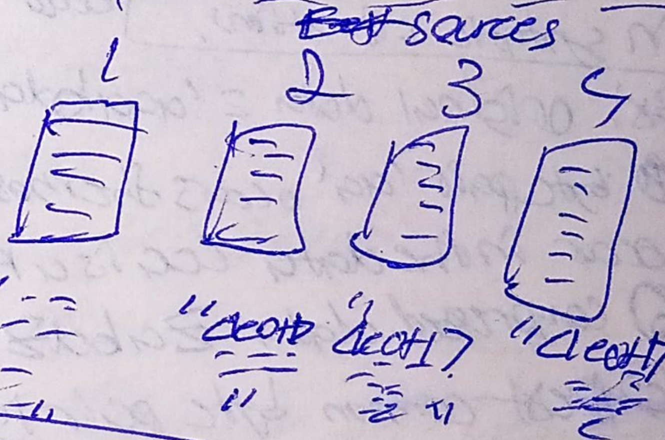
chased → 0
dog → 1
fox → 2
the → 3
<|unk|> → 783
<|endoftext|> → 784

Last 2 tokens

→ **Token IDs**

[3] [2] [0] [3] [1] [783] ...

Text sources



More details on <|endoftext|> <|eot|> = <|endoftext|>

* When working w/ multiple text sources, we add <|endoftext|> tokens between these texts
* These <|endoftext|> tokens act as markers, signaling the start or end of a particular segment
* This leads to more effective processing and understanding by the LLM

Byte pair encoding

* words themselves are broken into subwords, and these are the tokens
  (ch) (as) (ed); one possible subword breakdown
- used to train GPT-2, GPT-3, and OG Model for ChatGPT