Lecture 7: Tokenization from scratch

How to prepare input text data for training LLMs?

- **Step 1:** Splitting text into individual words and subword tokens.
- **Step 2:** Convert tokens into token IDs.
- **Step 3:** Encode token IDs into vector representations

Token IDs must be assigned in alphabetic order of the words.

Special Context Tokens

They're not part of "regular vocabulary" like words or subwords but serve as **control markers** that guide how models understand or process text.

1. `<unk>` (Unknown Token)

- **Purpose:** Represents words or symbols that are **not in the model's vocabulary**.
- **Why it exists:** No vocabulary can cover every possible word, typo, or rare token. Instead of crashing or ignoring unknown words, the model maps them to `<unk>`.
- **Example:**
  - Vocabulary: `["I", "love", "cats"]`
  - Input: `"I love quokkas"`
  - Tokenization: `["I", "love", "<unk>"]`
- **Implications:** The model won't know the exact word, but it can sometimes infer meaning from context. Too many `<unk>` tokens usually **hurt performance**, especially in specialized domains.

---

2. `<eos>` or `<endoftext>` (End-of-Sequence Token)

- **Purpose:** Marks the **end of a text sequence**.
- **Why it exists:** Models need to know when to **stop generating**. Without an end token, a model might keep producing text forever.
- **Example in generation:**
  - Prompt: `"Once upon a time, there was a dragon"`
  - Generated output: `" who loved painting.<endoftext>"`
  - Model sees `<endoftext>` → stops generation.
- **Other uses:** Often used in **sequence-to-sequence tasks** like translation, summarization, or dialogue models to indicate when a response is complete.

---

3. Other Common Special Tokens

- `<pad>` **(Padding Token):** Fills sequences to a **fixed length** for batch processing. Doesn't contribute to learning.
- `<bos>` **(Beginning of Sequence):** Marks the start of a sequence; useful for some autoregressive models.
- `<cls>` **(Classification Token):** In models like BERT, prepended to sequences to summarize the whole input for classification tasks.

---

4. Why They're Important

- **Control & structure:** They give models signals about **how to treat text**.
- **Handling unknowns:** `<unk>` prevents crashes and lets models generalize.

- **Text generation:** `<eos>` ensures **finite outputs**, crucial for inference.

The tokenizer used in GPT doesn’t use any of the tokens mentioned above, it only used the <endoftext> token.

GPT also doesn’t use <unk> token for unknown words. GPT model uses a **byte pair encoding tokenizer,** which breaks down words into subword units.