

chunking vs tokenization in llms in table form

Here's a table comparing **Chunking vs Tokenization in LLMs**:

Feature	Chunking	Tokenization
Definition	Dividing a large text into meaningful segments (chunks) for processing.	Converting text into smaller units (tokens) such as words, subwords, or characters.
Purpose	Helps manage long documents for vector storage, retrieval, and processing.	Prepares text for model input by breaking it into manageable pieces.
Granularity	Operates at sentence, paragraph, or document level.	Operates at word, subword, or character level.
Use Cases	RAG applications, document retrieval, embedding generation.	LLM text processing, sequence modeling, attention mechanisms.
Impact on LLMs	Affects how documents are stored, retrieved, and queried in a RAG pipeline.	Affects how an LLM understands and generates responses.
Common Methods	Fixed-length chunks, semantic splitting (e.g., by sentence or paragraph).	BPE (Byte Pair Encoding), WordPiece, Unigram, character-based tokenization.
Dependency on Model	Independent of the LLM's tokenizer; used for efficient information retrieval.	Directly tied to the LLM's tokenizer (e.g., GPT uses BPE).
Tools/Libraries	LangChain (for text chunking), NLTK, spaCy.	Hugging Face Tokenizers, SentencePiece, OpenAI Tokenizer.




Would you like a deeper dive into any specific aspect?



is chunking used in pre-training or fine tuning  
llm

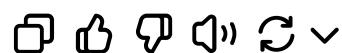
Chunking is **not typically used in the pre-training of an LLM**, but it is commonly applied during **fine-tuning** and **retrieval-augmented generation (RAG)**.

## Chunking in Pre-training vs Fine-tuning

Phase	Use of Chunking
Pre-training	 Not used. LLMs are trained on large-scale datasets using tokenization. The model learns from raw text processed in continuous token sequences rather than chunks.
Fine-tuning	 Used for handling long documents, summarization, retrieval-based tasks, and improving context management. Splitting large texts into chunks helps models process information more effectively.
RAG (Retrieval-Augmented Generation)	 Heavily used. Chunks of text are embedded and stored in a vector database for retrieval when responding to queries.

So, while **tokenization** is fundamental in **both pre-training and fine-tuning**, **chunking** is mainly useful in **fine-tuning**, **document retrieval**, and **RAG workflows**.

Would you like details on specific chunking strategies?



is tokenization used in pre-training or fine  
tuning llm or in RAG

## Tokenization in Pre-training, Fine-tuning, and RAG

Phase	Use of Tokenization
Pre-training	✅ Yes. Tokenization is fundamental. LLMs are trained on tokenized text rather than raw text. The model learns patterns based on token sequences.
Fine-tuning	✅ Yes. Tokenization is required because fine-tuning involves adapting the model on new datasets, which must be tokenized before being fed into the LLM.
RAG (Retrieval-Augmented Generation)	✅ Yes, but <b>indirectly</b> . Tokenization is used when querying the LLM, but chunking is more critical for storing and retrieving text in a vector database.

## Key Differences

- **Pre-training & Fine-tuning** → Tokenization is **mandatory** since LLMs process data as tokens.
- **RAG** → Tokenization is used when querying the LLM, but **chunking** is more important for document storage and retrieval.

Would you like a breakdown of tokenization methods (e.g., BPE, WordPiece)?

