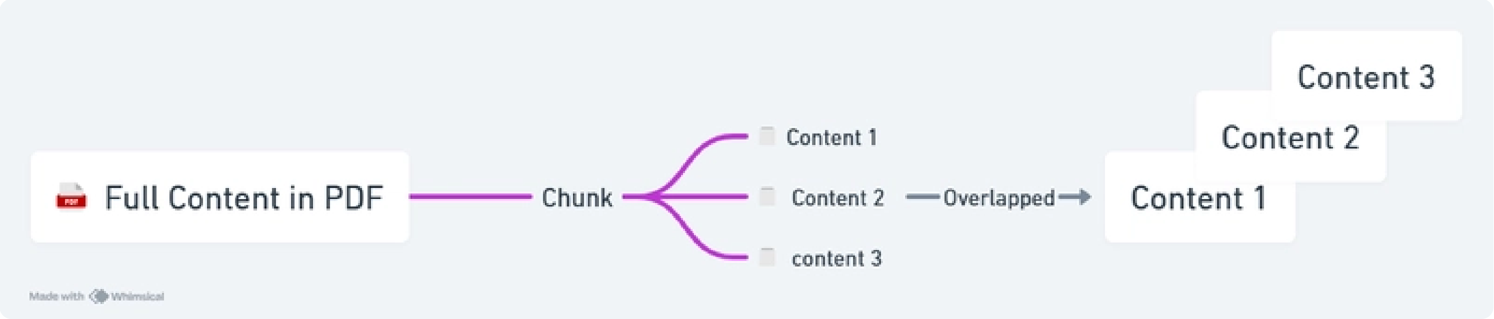
Taki
Posted on Dec 13, 2024 • Edited on Dec 14, 2024

12

What is Chunk Size and Chunk Overlap

#langchain #rag #ai #programming



What is Chunk Size?

The chunk size refers to **the maximum number of characters or tokens** allowed in a single chunk. For example, if the chunk size is 300, each chunk will have up to 300 characters.

Example:
Imagine this text:

> Chunking is the process of breaking a large piece of text into smaller, manageable pieces to make it easier to process and analyze.

if the **chunk size** is 40 characters:

- Chunk 1: "Chunking is the process of breaking a"
- Chunk 2: "large piece of the text into smaller,"
- Chunk 3: "manageable piece to make it easier"
- Chunk 4: "to process and analyze."

Notes: you can use tool to count of tokens Tokenizer or you can use tiktoken like that:

```
import { encoding_for_model } from "tiktoken";
const text = "Chunking is the process of breaking a large piece of text into smaller, mai
const tokenizer = encoding_for_model("gpt-3.5-turbo");
const tokens = tokenizer.encode(text);

console.log(`Total count of tokens: ${tokens.length}`);
```

What is Chunk Overlap?

Chunk Overlap refers to the **number of characters or tokens shared between consecutive chunks**. Overlapping ensures that important context is not lost when diving the text into smaller parts.

Example:
Using the same text:

> Chunking is the process of breaking a large piece of text into smaller, manageable pieces to make it easier to process and analyze.

With a **chunk size of 40 and an overlap of 10**:

- Chunk 1: "Chunking is the process of **breaking a**"
- Chunk 2: "**breaking a** large piece of the **text into**,"
- Chunk 3: "**text into**, manageable **piece**"
- Chunk 4: "**piece** to make it easier to **process**"
- Chunk 5: "**process** and analyze."

Here, "breaking a" is part of both Chunk 1 and Chunk 2, ensuring that no context is lost between chunks.

What should you use overlap?

- **Long sentences:** Prevents sentences from being split in a way that loses meaning.
- **Preserving context:** Ensures context is maintained for task like embedding or searching.

How to Chunk Text Programmatically (with Langchain):

```
import { RecursiveCharacterTextSplitter } from "langchain/text_splitter";

const text = "Chunking is the process of breaking a large piece of text into smaller, ma
const splitter = new RecursiveCharacterTextSplitter({
  chunkSize: 40,      // Maximum size of each chunk
  chunkOverlap: 10,   // Overlap between chunks
});

const chunks = await splitter.splitText(text);
console.log(chunks);
```

output:

```
[
  'Chunking is the process of breaking a',
  'a large piece of text into smaller,',
  'smaller, manageable pieces to make it',
  'make it easier to process and analyze.'
]
```

Why Chunking Matters

- For Embedding Models: Many models have a token limit, so chunking ensures the text fits.
- Improves Retrieval: Overlapping chunk helps capture text, making retrieval results more accurate.
- Scalability: Large texts are broken into smaller pieces, allowing efficient processing.

How to set Chunk Size and Chunk Overlap

- **Chunk Size:** Aim to fit within the model's token limit (e.g., Open AI models like *text-davinci-003* or *gpt-3.5-turbo* typically have token limits ranging from 4000 to 6000) but small enough for efficient processing. **Common values range from 200 to 500 tokens per chunk for embedding tasks.**
- **Chunk Overlap:** Generally, overlap is set to **10%-20% of the chunk size** to ensure continuity. This overlap ensures that the last few tokens of one chunk carry over to the next

chunk, preserving context for embedding or search tasks

Top comments (0)

Code of Conduct • Report abuse

Taki

Healing By Code

LOCATION
HCMC, Viet Nam

EDUCATION
University Of Science HCMC VN

JOINED
Aug 26, 2024

More from Taki

Developer can write unit test by cypress prompt
#typescript #nextjs #cypress #ai

Vite for React SPA

#webdev #javascript #programming #nextjs

AI Flashcard: NextJs (Basic)

#webdev #programming #ai #nextjs