Understanding RAG chunking through simple language, easy math, and playful AI demos!

20 Essential RAG Chunking Methods

Every AI Engineer Must Know

Sanjay N Kumar

Data scientist | AI ML Engineer | Statistician | Analytics Consultant

What is Chunking? 🧩

Chunking means **splitting large text into smaller parts (chunks)** so AI can understand better.

📖 Like dividing a long story into chapters!
Without chunking → AI forgets the beginning.
With chunking → AI remembers what matters.

Why Bad Chunking Fails 🚫

Bad chunking = AI confusion 🤯

- Breaks sentences halfway
- Loses topic meaning
- Gives wrong answers

Example:

If you cut ***"The cat sat on the mat."*** → into ***"The cat s" and "at on the mat"*** — meaning is lost!

Why Good Chunking Helps ✅

The forgotten realm awoke to a new dawn...
Heroes gathered from every corner...
The dragon's fire was finally quenched.

Good chunking = AI clarity 🌟

- Keeps sentences complete
- Maintains flow
- Gives correct answers

Example:

Reading one full paragraph at a time makes sense — not random half lines!

The 20 Chunking Techniques 📊

We’ll learn 20 common methods:

🧱 Fixed-size

✍️ Sentence

📄 Paragraph

🔁 Sliding window

🧠 Semantic

🔂 Recursive

🏗️ Structure-aware

💻 Code block

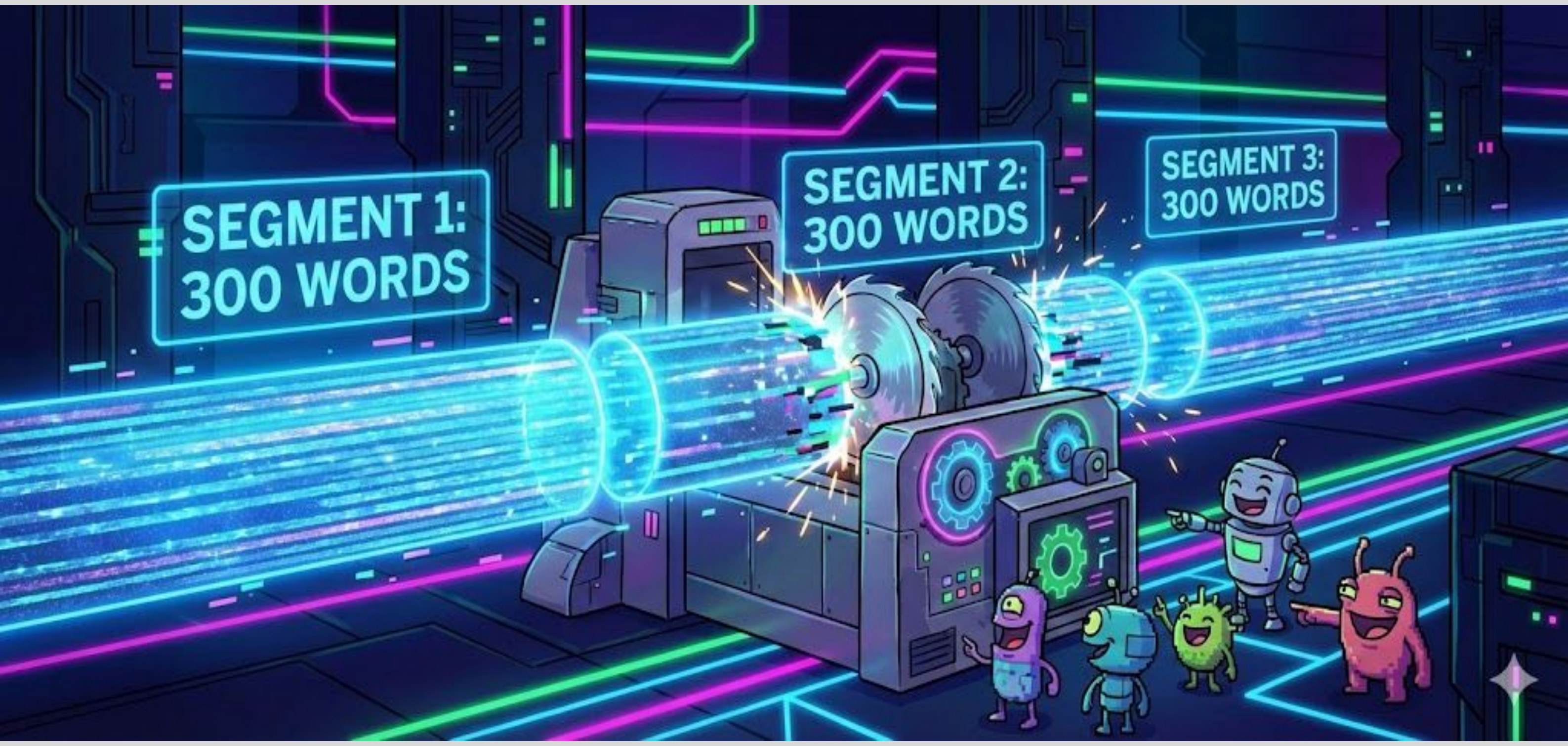
🎯 Query-aware dynamic

⚙️ Hybrid

The 20 Chunking Techniques 📊

- 📎 Context-enriched
- 🧮 Token-based
- 🤖 Agentic
- 📑 Page-based
- 📊 Table-aware
- 🖼️ Image-associated
- ⏱️ Event-based
- 📚 Citation-aware
- 🪜 Hierarchical
- 🎤 Audio/Text alignment

1. Fixed-Size Chunking 🧱



Text is split into **equal-sized parts** (like every 300 words).

✅ Simple, fast
❌ May break meaning mid-sentence

Example:
A 900-word story → split into 3 equal 300-word parts.

2. Sentence-Based Chunking ✍️

Each chunk = one full sentence.

✅ Keeps meaning complete

❌ May lose wider context between sentences

Example:

“Dogs bark.” → Chunk 1

“Cats meow.” → Chunk 2

3. Paragraph-Based Chunking 📄

Splits text by paragraph breaks.

✅ Natural separation of ideas

❌ Long paragraphs may exceed limits

Example:

Paragraph 1: “Plants make food.” 🌿

Paragraph 2: “Animals eat plants.” 🐄

4. Sliding Window Chunking 🔁

Chunks overlap slightly to keep flow.

✅ Keeps context

❌ Repeats some text

Example:

Chunk 1: Lines 1–5

Chunk 2: Lines 4–8 (overlaps 4–5)

5. Semantic Chunking 🧠

AI splits text by **meaning or topic**.

✅ Very accurate

❌ Slower

Example:

Text about “Earth 🌍” and “Mars 🔴” → becomes two chunks by topic.

6. Recursive Chunking 🔂

THE GAME TOME
CHAPTER 5: THE QUEST BEGINS
PARAGRAPH BLOCK: Find the pixel-art lanterns.
SENTENCE: THE MAGIC LIVES IN THE ...
HEALTH
SCORE
25
30,000
Almost there!
Where's the save point?
Where's the save point?

Splits large text step by step:

Document → Section → Paragraph → Sentence.

✅ Flexible

❌ Needs document structure

Example:

Book → Chapter → Paragraph → Line

7. Structure-Aware Chunking 🏗️

Uses **document structure** like headings, lists, or tags.

✅ Best for websites or reports
❌ Needs clean formatting

Example:
HTML <h1>, <h2>, <p> → become separate chunks.

8. Code Block Chunking 💻

Splits text by **functions, classes, or methods** in code.

✅ Keeps logic complete

❌ Hard for mixed text+code docs

Example:

Chunk 1: `def add_numbers()`

Chunk 2: `def subtract_numbers()`

9. Query-Aware Dynamic Chunking 🎯

AI changes chunking style **based on your question**.

✅ Most relevant results

❌ Complex to build

Example:

If you ask "loan process," it reads only loan-related chunks.

10. Hybrid Chunking ⚙️

Combines two or more methods.

✅ Best of both worlds

❌ Setup is complex

Example:

Use *semantic + sliding window* → meaning + context together.

11. Context-Enriched Chunking 📎

Adds extra info like **title or source** to each chunk.

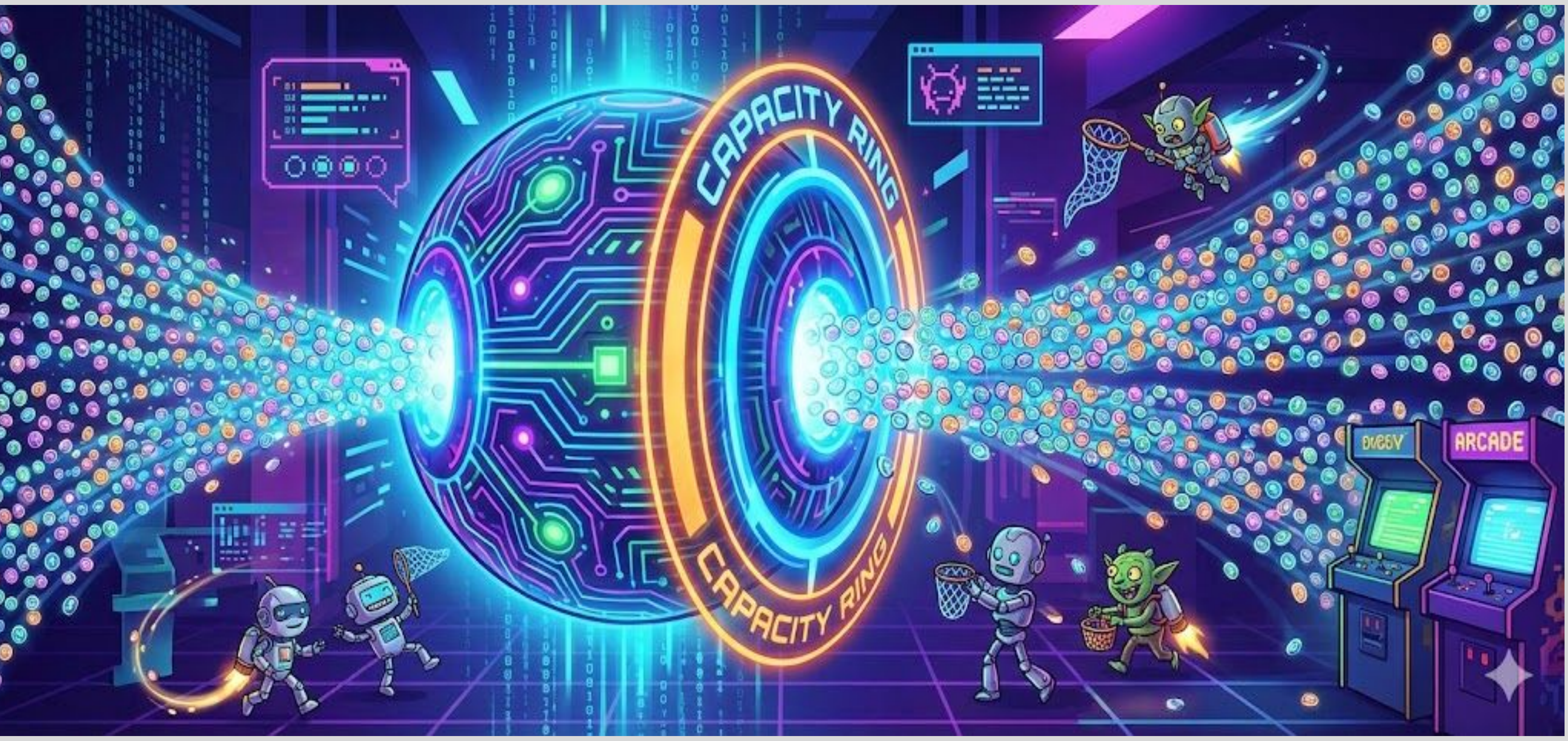
✅ Improves traceability

❌ Uses more tokens

Example:

“Chapter 2 – Photosynthesis 🌿: Plants make food from light.”

12. Token-Based Chunking 🧮



AI reads “tokens,” not words.

Splits exactly by token limit (like 512 or 1024).

✅ Prevents overflow

❌ May break grammar

Example:

Model fits 1000 tokens → each chunk = 1000-token portion.

13. Agentic Chunking 🤖

AI itself decides where to cut based on structure or meaning.

✅ Smart and adaptive
❌ Computationally heavy

Example:
LLM splits a 20-page report into logical 5-topic groups.

14. Page-Based Chunking 📑

Each **page of a PDF or scan** is one chunk.

✅ Keeps layout and numbering
❌ Not meaning-based

Example:

Page 1 → Chunk 1

Page 2 → Chunk 2

15. Table-Aware Chunking 📊

Splits tables logically by **rows, columns, or headers**.

✅ Great for data-heavy files

❌ Needs pattern detection

Example:

Chunk 1: Rows 1–5

Chunk 2: Rows 6–10

16. Image-Associated Chunking 🖼️

Pairs images with nearby text or captions.

✅ Perfect for multimodal RAG

❌ Requires OCR or captioning

Example:

Image of a volcano 🌋 + text “Mount Fuji eruption” → 1 chunk

17. Event-Based Chunking ⏱️

Splits text by **time or event markers** (like timestamps).

✅ Keeps chronological flow
❌ Needs structured logs or transcripts

Example:

[10:00] Meeting start → Chunk 1

[10:30] Discussion → Chunk 2

18. Citation-Aware Chunking 📚

ANCIENT MAGIC THEORY [1]
DRAGON PHYSIOLOGY [5]
[1] Merlin, A. "The Big Book of Spells" (1023)
[5] Smaug, B. "Why I Breathe Fire" (3002)

Keeps **references and citations** with the related text.

✅ Useful for legal or research papers

❌ Increases chunk size

Example:

“Quantum theory [Einstein, 1905] explains energy behavior.”

19. Hierarchical Chunking 🪜

Creates multi-level chunks:

Sentence → Paragraph → Section → Document.

✅ Supports flexible retrieval

❌ Complex to manage

Example:

You can search at “section level” or “paragraph level.”

20. Audio/Text Alignment Chunking 🎤

Used in speech data — aligns chunks with timestamps or speakers.

✅ Keeps voice context

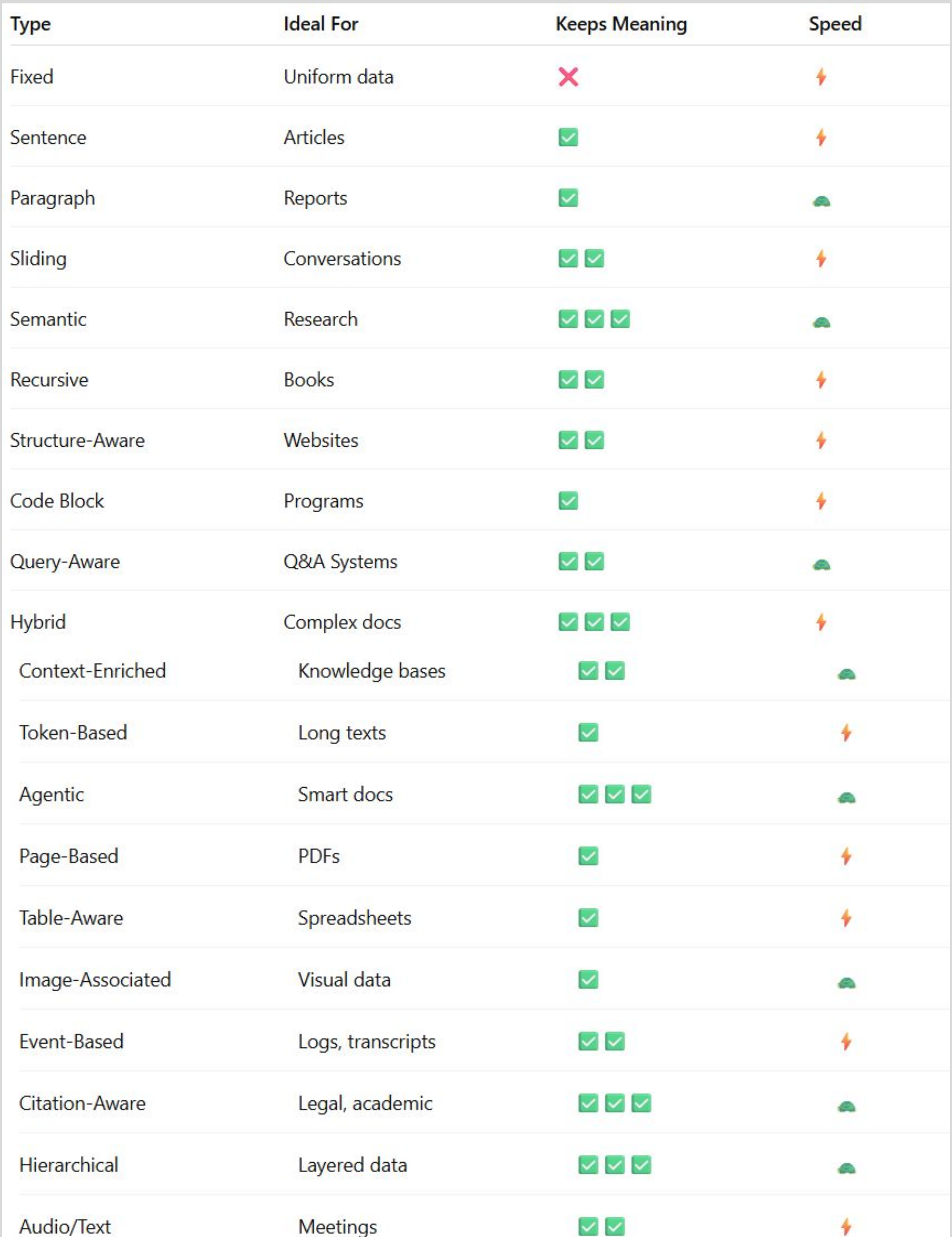
❌ Needs audio timing

Example:

Speaker 1 (00:00–01:00) → Chunk 1

Speaker 2 (01:00–02:00) → Chunk 2

Summary Table 🧩



Type	Ideal For	Keeps Meaning	Speed
Fixed	Uniform data	❌	⚡
Sentence	Articles	✅	⚡
Paragraph	Reports	✅	🐢
Sliding	Conversations	✅✅	⚡
Semantic	Research	✅✅✅	🐢
Recursive	Books	✅✅	⚡
Structure-Aware	Websites	✅✅	⚡
Code Block	Programs	✅	⚡
Query-Aware	Q&A Systems	✅✅	🐢
Hybrid	Complex docs	✅✅✅	⚡
Context-Enriched	Knowledge bases	✅✅	🐢
Token-Based	Long texts	✅	⚡
Agentic	Smart docs	✅✅✅	🐢
Page-Based	PDFs	✅	⚡
Table-Aware	Spreadsheets	✅	⚡
Image-Associated	Visual data	✅	🐢
Event-Based	Logs, transcripts	✅✅	⚡
Citation-Aware	Legal, academic	✅✅✅	🐢
Hierarchical	Layered data	✅✅✅	🐢
Audio/Text	Meetings	✅✅	⚡

Final Takeaway 🎯

🧠 Chunking is how you teach your AI *where to pause and think*.

If you chunk wrongly — AI confuses facts.

If you chunk wisely — AI becomes accurate, calm, and smart. 🌟

Fuel Your RAG. Shape the Future 🔮

Every chunk you create writes tomorrow's intelligent answers.

💡 *Build once. Retrieve forever.*

Ready to train smarter? Let's connect and optimize the future!

Sanjay N Kumar
Data scientist | AI ML Engineer | Statistician | Analytics Consultant

https://www.linkedin.com/in/sanjaytheanalyst360/

sanjaytheanalyst360@gmail.com