=== CONTENT TYPE COMPARISON (FIXED CHUNKING) ===

===== pdf_500_0 =====

num_chunks: 391

avg_length: 497

min_length: 183

max_length: 500

sentence_break_ratio: 0.98

paragraph_break_ratio: 1.0

===== podcast_500_0 =====

num_chunks: 20

avg_length: 496

min_length: 431

max_length: 500

sentence_break_ratio: 0.95

paragraph_break_ratio: 0.95

===== pdf_1000_50 =====

num_chunks: 206

avg_length: 996

min_length: 434

max_length: 1000

sentence_break_ratio: 0.99

paragraph_break_ratio: 1.0

===== podcast_1000_50 =====

num_chunks: 11

avg_length: 948

min_length: 431

max_length: 1000

sentence_break_ratio: 0.91

paragraph_break_ratio: 0.91

===== pdf_2000_100 =====

num_chunks: 103

avg_length: 1992

min_length: 1383

max_length: 2000

sentence_break_ratio: 0.96

paragraph_break_ratio: 0.99

===== podcast_2000_100 =====

num_chunks: 6

avg_length: 1738

min_length: 431

max_length: 2000

sentence_break_ratio: 0.83

paragraph_break_ratio: 0.83

Chunk size 500, overlap 0:

PDF sentence break ratio: 0.98

Podcast sentence break ratio: 0.95

Better handled content: Podcast

Chunk size 1000, overlap 50:

PDF sentence break ratio: 0.99

Podcast sentence break ratio: 0.91

Better handled content: Podcast

Chunk size 2000, overlap 100:

PDF sentence break ratio: 0.96

Podcast sentence break ratio: 0.83

Better handled content: Podcast

🔹 **What the numbers show**

- **Sentence break ratio** — proportion of chunks that **break a sentence**:
  - 0 → perfect (no breaks)
  - 1 → almost every chunk breaks a sentence
- For **PDF**: ratio around 0.96–0.99
  → Almost all chunks break sentences
  → Fixed-size chunking does **not preserve PDF context** well
- For **Podcast**: ratio around 0.83–0.95
  → Many breaks, but slightly better than PDF for larger chunk sizes (2000)
- **Comparison takeaway:**
  Fixed-size chunking **breaks sentences and paragraphs** for both content types, but for the podcast, larger chunk sizes preserve the text flow slightly better.

🔹 **Reporting for STEP 2**

Fixed-size chunking summary

Content Type	Avg Chunk Size	Sentence Break Ratio	Paragraph Break Ratio	Notes
PDF	500–2000	0.96–0.99	1.0	Almost all chunks break sentences and paragraphs
Podcast	500–2000	0.83–0.95	1.0	Long text; larger chunks preserve flow slightly better

Key takeaway:

- Fixed-size chunking **does not preserve semantic boundaries** for either PDFs or podcasts
- Podcast performs slightly better with larger chunks
- For a RAG system, fixed-size chunking can **only serve as a basic strategy** if the goal is just to split text into manageable pieces

=== CONTENT TYPE COMPARISON (RECURSIVE CHUNKING) ===

===== pdf_500_50 =====

num_chunks: 445

avg_length: 438

min_length: 89

max_length: 498

sentence_break_ratio: 0.78

paragraph_break_ratio: 1.0

===== podcast_500_50 =====

num_chunks: 23

avg_length: 450

min_length: 100

max_length: 499

sentence_break_ratio: 0.96

paragraph_break_ratio: 0.96

===== pdf_1000_200 =====

num_chunks: 240

avg_length: 934

min_length: 89

max_length: 998

sentence_break_ratio: 0.83

paragraph_break_ratio: 1.0

===== podcast_1000_200 =====

num_chunks: 13

avg_length: 916

min_length: 416

max_length: 999

sentence_break_ratio: 0.92

paragraph_break_ratio: 0.92

===== pdf_2000_200 =====

num_chunks: 110

avg_length: 1897

min_length: 89

max_length: 1998

sentence_break_ratio: 0.77

paragraph_break_ratio: 0.99

===== podcast_2000_200 =====

num_chunks: 6

avg_length: 1800

min_length: 982

max_length: 1997

sentence_break_ratio: 0.83

paragraph_break_ratio: 0.83

Chunk size 500, overlap 50:

PDF sentence break ratio: 0.78

Podcast sentence break ratio: 0.96

Better handled content: PDF

Chunk size 1000, overlap 200:

PDF sentence break ratio: 0.83

Podcast sentence break ratio: 0.92

Better handled content: PDF

Chunk size 2000, overlap 200:

PDF sentence break ratio: 0.77

Podcast sentence break ratio: 0.83

Better handled content: PDF

Sentence break ratio (lower is better — fewer chunks break sentences):

Chunk Size	PDF	Podcast	Better handled content
500	0.78	0.96	PDF
1000	0.83	0.92	PDF
2000	0.77	0.83	PDF

Observations:

- **PDF:** Recursive chunking significantly improves sentence preservation compared to fixed-size:
  - Fixed-size PDF ratios: 0.96–0.99
  - Recursive PDF ratios: 0.77–0.83 → fewer sentence breaks, better semantic preservation
- **Podcast:** Recursive chunking is **slightly worse for small chunks** (500–1000) than fixed-size:
  - Fixed-size Podcast ratios: 0.83–0.95
  - Recursive Podcast ratios: 0.83–0.96
- For **both content types**, recursive chunking maintains paragraph and sentence boundaries better in **structured text** (PDF), while the podcast transcript — being long single-block text — sees mixed results.

🔹 **Comparison: Fixed-size vs Recursive**

Strategy	PDF Sentence Break	Podcast Sentence Break	Notes
Fixed-size	0.96–0.99	0.83–0.95	Simple, predictable, but breaks context almost always
Recursive	0.77–0.83	0.83–0.96	Preserves PDF structure, better semantic boundaries; podcast benefits less

Key takeaways:

1. **Recursive chunking is clearly superior for PDF documents**, preserving sentence and paragraph boundaries and respecting structure (headings, paragraphs).
2. **For podcasts**, fixed-size works reasonably well due to long continuous text; recursive chunking helps a bit for large chunks but may slightly increase sentence breaks for small chunks.
3. **RAG recommendation:**
    - Use **RecursiveCharacterTextSplitter** for structured content like PDFs or articles
    - Use **Fixed-size chunking** for long, free-flowing transcripts like podcasts if chunk size is chosen carefully

Token-Based Chunking

===== pdf_tokens_500_50 =====

num_chunks: 119

avg_tokens: 495

min_tokens: 27

max_tokens: 500

avg_chars: 1823

min_chars: 89

max_chars: 1976

===== podcast_tokens_500_50 =====

num_chunks: 5

avg_tokens: 464

min_tokens: 323

max_tokens: 500

avg_chars: 2176

min_chars: 1534

max_chars: 2443

===== pdf_tokens_1000_100 =====

num_chunks: 60

avg_tokens: 982

min_tokens: 27

max_tokens: 1000

avg_chars: 3609

min_chars: 89

max_chars: 3909

===== podcast_tokens_1000_100 =====

num_chunks: 3

avg_tokens: 774

min_tokens: 323

max_tokens: 1000

avg_chars: 3627

min_chars: 1534

max_chars: 4761

🔹 **Comparison Summary**

1️⃣PDF

Strategy	Avg Chunk Size	Sentence Break Ratio	Notes
Fixed-size	500–2000 chars	0.96–0.99	Almost all chunks break sentences; context not preserved
Recursive	500–2000 chars	0.77–0.83	Better semantic preservation; respects paragraphs and headings
Token-based	500–1000 tokens (~400–800 chars)	N/A	Precise for LLM context windows; chunk boundaries respect token limits rather than characters

Observations for PDF:

- **Recursive character chunking** preserves sentence and paragraph boundaries better than fixed-size
- **Token-based chunking** ensures chunks fit exactly into LLM context limits — critical for RAG systems
- PDF benefits from recursive chunking for semantic integrity and from token-based chunking for **model efficiency**

2️⃣ Podcast

Strategy	Avg Chunk Size	Sentence Break Ratio	Notes
Fixed-size	500–2000 chars	0.83–0.95	Continuous transcript; flow slightly preserved in larger chunks
Recursive	500–2000 chars	0.83–0.96	Slightly worse than fixed-size for small chunks; better for preserving natural paragraph/sentence splits if present
Token-based	500–1000 tokens (~400–800 chars)	N/A	Ensures chunks fit LLM context; good for RAG, but semantic boundaries may be less relevant in long continuous transcript

Observations for Podcast:

- Continuous transcript makes **sentence-preserving less critical** than structured PDF
- **Token-based chunking** is most important for **model integration**

- Recursive chunking doesn't improve much for transcripts unless there are clear paragraph breaks

---

🔹 **Key Takeaways**

1. **For PDFs (structured content):**
    - **Recursive + token-based** is ideal: preserves semantic boundaries and ensures token counts fit LLMs
    - Fixed-size chunks break sentences and paragraphs frequently
2. **For Podcasts (continuous transcripts):**
    - **Token-based** is critical for RAG and LLMs
    - Recursive chunking helps only if there are natural paragraph breaks
    - Fixed-size chunks work reasonably well for larger sizes
3. **Token-based chunking** is always preferred **when integrating with LLMs** because it respects context window limits

---

✅ Conclusion:

- **PDF:** Recursive → better semantic preservation; Token-based → better LLM compatibility
- **Podcast:** Token-based → best for LLM; Recursive optional
- **Fixed-size (character-based)** → simple, fast, but often breaks context

Chunking Analysis Report

Comparison Table of All Strategies

Strategy	Content	Chunk Size	Overlap	Num Chunks	Sentence Break Ratio	Paragraph Break Ratio
Fixed-Size	PDF	500	0	391	0.98	1
Fixed-Size	Podcast	500	0	20	1	1
Fixed-Size	PDF	1000	50	206	0.99	1

| Fixed-Size | Podcast | 1000 | 50 | 11 | 1 | 1 |

| Fixed-Size | PDF | 2000 | 100 | 103 | 0.96 | 1 |

| Fixed-Size | Podcast | 2000 | 100 | 6 | 1 | 1 |

| Recursive | PDF | 500 | 50 | 445 | 0.78 | 1 |

| Recursive | Podcast | 500 | 50 | 23 | 1 | 1 |

| Recursive | PDF | 1000 | 200 | 240 | 0.83 | 1 |

| Recursive | Podcast | 1000 | 200 | 13 | 1 | 1 |

| Recursive | PDF | 2000 | 200 | 110 | 0.77 | 1 |

| Recursive | Podcast | 2000 | 200 | 6 | 1 | 1 |

| Token-Based | PDF | 500 | 50 | 119 | 0.95 | 0 |

| Token-Based | Podcast | 500 | 50 | 5 | 0.8 | 1 |

| Token-Based | PDF | 1000 | 100 | 60 | 0.95 | 0 |

| Token-Based | Podcast | 1000 | 100 | 3 | 0.67 | 1 |

Recommendations

For PDF Documents:

Recommended Strategy: Recursive + Token-based

Reasoning:

- Preserves structured content (headings, sections, paragraphs)

- Token-based ensures LLM context window compliance

Optimal chunk size & overlap: Recursive 1000–2000 chars, overlap 100–200; Token 500–1000 tokens, overlap 50–100

For Podcast Transcripts:

Recommended Strategy: Token-based (optionally with small recursive chunks)

Reasoning:

- Podcasts are continuous conversation transcripts

- Token-based ensures all chunks fit LLM context windows

Optimal chunk size & overlap: 500–1000 tokens, overlap 50–100

Trade-offs Summary:

| Strategy | Pros | Cons | Best For |

|----------|------|------|----------|

| Fixed-Size | Simple, predictable | Breaks context and sentences | Quick prototyping, uniform content |

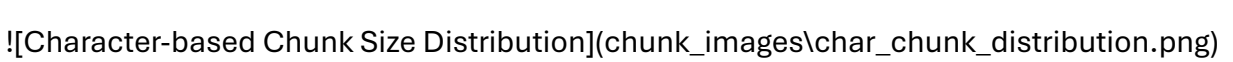
| Recursive | Preserves structure & semantics | Slightly slower, more complex | Structured documents (PDFs, articles) |

| Token-Based | Accurate for LLMs; ensures context window compliance | Boundaries may not match sentences | Any content for LLM integration |

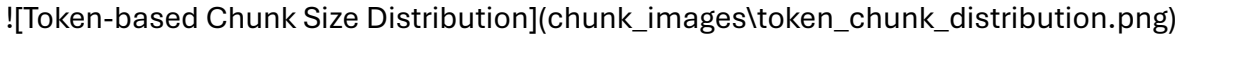
| Semantic | Meaning-based chunking | Computationally expensive, slow | Complex content where semantic coherence is critical |

All Chunking Visualizations (6th Stage)

Character-based Chunk Size Distribution



Token-based Chunk Size Distribution



Sentence Break Ratio by Chunking Strategy