AGENTIC AI LAB

(CSCR3215)

BACHELOR OF TECHNOLOGY

In

Computer Science and Engineering

Department of Computer Science and Engineering

School of Computing Science & Engineering

Sharda University, Greater Noida

Jan-June 2026

Objective:

To demonstrate different levels of text splitting techniques used in Natural Language Processing (NLP) for efficient text chunking, preprocessing, and preparation for downstream tasks such as embeddings, search, and large language models.

Overview of the Working

The notebook implements **five progressive levels of text splitting**, starting from basic splitting to advanced, context-aware chunking. Each level improves how text is divided while preserving meaning and usability.

Level 1: Character-Based Text Splitting

- Text is split purely based on a fixed number of characters.
- This is the simplest method and does not consider sentence or word boundaries.
- **Output:** Uniform chunks of text with possible breaks in words or sentences.
- **Use Case:** Quick splitting when structure is not important.

Level 2: Recursive Character Text Splitting

- Text is split recursively using a hierarchy of separators (paragraph → sentence → word → character).
- Prevents abrupt breaks when possible.
- **Output:** More meaningful chunks compared to Level 1.
- **Advantage:** Maintains partial semantic structure.

Level 3: Sentence-Based Text Splitting

- Text is divided at sentence boundaries.
- Each chunk contains complete sentences.
- **Output:** Readable and logically consistent text chunks.
- **Use Case:** NLP tasks like summarization and question answering.

Level 4: Token-Based Text Splitting

- Splitting is based on token count instead of characters.

- Ensures chunks stay within model token limits.
- **Output:** Token-safe chunks compatible with LLMs.
- **Importance:** Prevents token overflow errors in transformer models.

Level 5: Semantic / Context-Aware Text Splitting

- Splitting is done while preserving semantic meaning.
- Related sentences are grouped together.
- **Output:** Contextually meaningful chunks with higher relevance.
- **Use Case:** Vector databases, retrieval-augmented generation (RAG), embeddings.

Final Observation

From the implementation and results of the five levels of text splitting, it is observed that the quality and usefulness of text chunks improve significantly as we move from basic to advanced splitting techniques. Character-based splitting is simple but often breaks words and sentences, leading to loss of meaning. Recursive character splitting performs better by attempting to preserve structure, but still lacks full contextual awareness. Sentence-based splitting produces more readable and logically complete chunks, making it suitable for many NLP tasks. Token-based splitting is especially effective for working with large language models, as it ensures that text chunks remain within token limits. The semantic or context-aware text splitting method provides the best results by preserving meaning and relationships between sentences, making it highly suitable for advanced applications such as embeddings, retrieval systems, and conversational AI.

Conclusion

The notebook successfully demonstrates how text splitting evolves from simple character division to advanced semantic chunking. Higher-level splitting methods significantly improve contextual integrity, making them suitable for modern NLP applications such as search engines, chatbots, and large language models.