Agentic AI Lab

(CSCR 3215)

BACHELOR OF TECHNOLOGY

In

Computer Science and Engineering

Submitted by:

Mayank Kumar

(2023542006) CSE-F G2

Submitted to: Mr. Ayush Singh
Assistant Professor

Department of Computer Science and Engineering

School of Computing Science & Engineering

Sharda University, Greater Noida

Jan-June 2026

Lab 2: Five Levels of Text Splitting for Multimodal Applications

1. Introduction

Text splitting is a critical preprocessing step in modern natural language processing and multimodal systems. Large documents, images with embedded text, and multimedia data often need to be segmented into smaller, meaningful chunks before further processing. Proper text segmentation improves information retrieval, semantic understanding, and downstream tasks such as summarization, question answering, and retrieval-augmented generation (RAG).

This lab explores five hierarchical levels of text splitting designed for multimodal data processing.

2. Objective

The objectives of this lab are:

• To understand the importance of text splitting in NLP and multimodal systems.

• To implement and analyze five different levels of text splitting.

• To study how different splitting strategies affect contextual understanding.

• To prepare text for efficient embedding and retrieval.

3. Levels of Text Splitting

The notebook demonstrates the following five levels:

1. Character-Level Splitting

Text is divided into fixed-size character chunks. This method ensures uniform chunk size but may break semantic meaning.

2. Word-Level Splitting

Text is split based on words, maintaining better readability than character-level splitting.

3. Sentence-Level Splitting

Text is segmented into complete sentences, preserving grammatical structure and semantic clarity.

4. Paragraph-Level Splitting

Paragraph-based splitting maintains contextual continuity and is suitable for long-form documents.

5. Semantic / Recursive Splitting

Advanced splitting that dynamically adjusts chunk size based on semantic boundaries and context windows, commonly used in RAG pipelines.

4. Methodology

The following approach is used in the implementation:

- Raw text input is loaded into the system.

- Each splitting strategy is applied sequentially.

- Outputs are compared to understand trade-offs between context size and semantic coherence.

- The suitability of each method for multimodal and LLM-based applications is analyzed.

5. Implementation Details

- Programming Language: Python
- Libraries Used: LangChain / NLP utilities
- Execution Environment: Jupyter Notebook / Google Colab
- Use Case: Preparing text for embeddings, retrieval, and multimodal pipelines

The implementation is modular and allows easy experimentation with chunk size and overlap.

6. Results and Observations

Key observations from this experiment include:

- Smaller chunks improve retrieval accuracy but may lose context.
- Larger chunks preserve meaning but can exceed token limits.
- Sentence and semantic-level splitting provide the best balance between context preservation and processing efficiency.
- Recursive splitting is particularly effective for multimodal and LLM-based applications.

7. Conclusion

This lab demonstrates the importance of structured text splitting in modern AI systems. The five-level approach provides flexibility depending on the application requirements. Semantic and recursive splitting methods are especially valuable for multimodal AI, retrieval-augmented generation, and large language model workflows. Proper text splitting significantly enhances model performance and system scalability.