Agentic AI Lab

(CSCR 3215)

B.TECH 3rd YEAR

SEMESTER: 6th

SESSION: 2025-2026

Submitted By:

Bhavishya Bhardwaj

(2023498559)

SECTION: H (G2)

Submitted To

Mr. Ayush
Kumar Singh
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SHARDA SCHOOL OF ENGINEERING & TECHNOLOGY

SHARDA UNIVERSITY, GREATER NOIDA

Lab Report: 5 Levels of Text Splitting

Aim

To study and implement different **levels of text splitting techniques** used in Natural Language Processing (NLP) for efficient text processing, chunking, and downstream tasks such as retrieval, summarization, and question answering.

Objective

- Understand why text splitting is required in NLP
- Implement multiple text splitting strategies
- Analyze the effect of different splitting levels
- Compare structured vs unstructured splitting

Software & Tools Used

- **Programming Language:** Python
- **Platform:** Jupyter Notebook / Google Colab
- **Libraries:**
    - langchain
    - textwrap
    - re
    - nltk (if used)
    - os

Theory

Large text documents cannot be processed directly by language models due to **token limits**. Text splitting divides large text into smaller, manageable chunks while preserving meaning and structure.

Different splitting levels are used based on:

- Context retention
- Semantic coherence
- Model constraints

Methodology / Levels of Text Splitting

Level 1: Character-Based Text Splitting

- Splits text based on a fixed number of characters.
- Simple but may break sentences or words.

Use Case: Basic chunking when structure is not important.

Level 2: Recursive Character Text Splitting

- Splits text hierarchically using separators like paragraphs, sentences, and characters.
- Preserves structure better than simple character splitting.

Use Case: Long documents with mixed formatting.

Level 3: Token-Based Text Splitting

- Splits text based on token count rather than characters.
- Ensures compatibility with language model token limits.

Use Case: LLM-based applications.

Level 4: Sentence-Based Text Splitting

- Splits text into complete sentences.
- Maintains semantic integrity.

Use Case: Question answering and summarization tasks.

Level 5: Semantic / Document-Based Splitting

- Splits text using document structure such as headings, sections, or meaning.
- Provides highest contextual relevance.

Use Case: Retrieval-Augmented Generation (RAG) systems.

Code Explanation

- Input text is loaded from a file or variable.
- Different text splitters are applied sequentially.
- Output chunks are printed and analyzed.

- Chunk size and overlap are adjusted to optimize results.

Results

- Character splitting produces fast but less meaningful chunks.
- Recursive splitting maintains better structure.
- Token and sentence-based splitting produce context-aware chunks.
- Semantic splitting provides the best retrieval performance.

Applications

- Chatbots
- Document retrieval systems
- Text summarization
- Question answering systems
- LLM-based applications

Advantages

- Improves efficiency and accuracy
- Prevents token overflow
- Enhances contextual understanding

Limitations

- Improper chunk size may reduce performance
- Semantic splitting requires additional computation

Conclusion

This experiment demonstrated five different levels of text splitting techniques. Advanced splitting methods preserve semantic meaning and significantly improve performance in NLP and LLM-based applications.

Future Scope

- Adaptive chunk sizing
- Multilingual text splitting
- Integration with vector databases
