SHARDA UNIVERSITY

Beyond Boundaries

AGENTIC AI LAB

CSCR3215

B.Tech. (CSE)-VI Semester

School of Engineering & Technology
Department of Computer Science & Engineering

Submitted to:
Mr. Ayush Singh

Submitted by:
Sumit Shrestha
2023850898

1. Aim of the Experiment

The aim of this experiment is to study, implement, and compare different levels of text splitting techniques used in modern Natural Language Processing (NLP) and Agentic AI systems. The lab focuses on understanding why text splitting is required and how different strategies affect downstream tasks such as retrieval, embedding, and reasoning.

2. Problem Statement

Large Language Models (LLMs) cannot process very large documents directly due to context-length limitations. When working with long documents such as PDFs, books, or web pages, the text must be divided into smaller chunks.

The challenge is:

How to split text without losing semantic meaning

How to balance chunk size and context preservation

How different splitting strategies impact performance

This lab explores five progressive levels of text splitting, from simple to advanced.

3. Theory / Background

3.1 What is Text Splitting?

Text splitting is the process of dividing a large body of text into smaller, manageable chunks that can be:

Embedded

Stored in vector databases

Retrieved efficiently

Processed by LLMs

Poor splitting can break semantic continuity, while good splitting preserves meaning across chunks.

3.2 Importance in Agentic AI

In Agentic AI systems, text splitting is critical for:

Retrieval-Augmented Generation (RAG)

Long-term memory storage

Tool-based reasoning

Document understanding agents

Effective agents rely on well-structured chunks to reason accurately over documents.

4. Levels of Text Splitting

This experiment demonstrates five levels of text splitting, increasing in sophistication.

5. Level 1: Character-Based Splitting

Description

The simplest form of text splitting divides text based on a fixed number of characters.

Characteristics

Easy to implement

No understanding of language structure

High risk of breaking sentences and words

Use Case

Used mainly for quick experimentation or when semantic accuracy is not critical.

6. Level 2: Word-Based Splitting

Description

Text is split based on word count rather than raw characters.

Characteristics

Prevents word breakage

Still ignores sentence and paragraph boundaries

Slightly better semantic preservation

7. Level 3: Sentence-Based Splitting

Description

Text is divided using sentence boundaries, typically identified by punctuation.

Characteristics

Preserves grammatical structure

Maintains semantic coherence

Suitable for most NLP tasks

Limitation

Very long sentences may still exceed token limits.

8. Level 4: Recursive Text Splitting

Description

Recursive splitting applies multiple strategies hierarchically:

Paragraphs

Sentences

Words

Characters
The text is split only when size constraints are violated.
Advantages
Preserves maximum context
Highly flexible
Widely used in RAG pipelines

9. Level 5: Semantic Text Splitting

Description
Semantic splitting uses embeddings or topic similarity to split text at meaningful boundaries.
Characteristics
Context-aware
Best semantic preservation
Computationally expensive
Application
Used in advanced agentic systems and knowledge-based retrieval systems.

10. Implementation Overview

Importing Required Libraries

```
from langchain.text_splitter import (
   CharacterTextSplitter,
   RecursiveCharacterTextSplitter
)
```

These utilities provide ready-made implementations for different splitting strategies.

Example Text Input

```
text = """
Large documents are difficult for language models to process...
"""
```

A long document is used as input to demonstrate splitting behavior.

Applying Different Splitters

```
splitter = CharacterTextSplitter(chunk_size=100, chunk_overlap=20)
chunks = splitter.split_text(text)
```

Different splitters are applied with varying parameters to observe their effects.

11. Observations and Analysis

Smaller chunks improve retrievability but reduce context
Larger chunks preserve meaning but risk token overflow
Recursive splitting provides the best balance
Semantic splitting produces the most meaningful chunks

12. Results

The experiment demonstrates that no single text splitting method is optimal for all tasks. The choice depends on:
Document size
Task complexity
Model context window

13. Conclusion

This lab provides a clear understanding of why text splitting is a foundational component of Agentic AI systems. By exploring five levels of splitting, the experiment highlights the trade-offs between simplicity, efficiency, and semantic accuracy.