Efficient Retrieval-Augmented Generation for Practical Analytics

Abstract

This report examines techniques for splitting unstructured documents into smaller, searchable units—“chunks”—to improve retrieval accuracy in Retrieval-Augmented Generation (RAG) pipelines. We evaluate fixed-size, overlapping, structure-aware, semantic, and recursive chunking strategies on a small corpus and discuss trade-offs among coherence, index size, and latency. Results indicate that structure-aware and recursive chunking improve answer grounding for mixed-length documents, while semantic chunking is most effective when topic boundaries are explicit but requires additional compute.

1. Introduction

Retrieval-Augmented Generation (RAG) pairs a retriever with a large language model (LLM) to answer questions grounded in external data. A persistent challenge is how to split documents into chunks that are both coherent and compact enough for efficient retrieval. Naive splitting by fixed length can fragment meaning; purely structural splitting may create oversized chunks. This paper explores chunking strategies that balance coherence with efficiency. We focus on simple heuristics that are practical for small teams and extend naturally to larger deployments.

2. Method

We compare five chunking strategies: - Fixed-size - Overlapping windows - Structure-aware splitting by headings - Semantic splitting via sentence embeddings - Recursive splitting that falls back from paragraphs to sentences when token limits are exceeded. We measure chunk size distributions, retrieval overlap with gold passages, and qualitative answer grounding.

3. Results

Fixed-size splitting is fast but often cuts through sentences. Overlap partially mitigates boundary loss while increasing index size. Structure-aware chunking aligns well with headings when documents are cleanly formatted. Semantic splitting groups related sentences effectively but is sensitive to threshold choice. Recursive splitting produced consistent chunk sizes across varied paragraph lengths and improved downstream prompt fit.

4. Discussion

For heterogeneous documents, a hybrid of structure-aware and recursive approaches offered the best balance. When topic transitions are abrupt, semantic splitting improves precision but adds embedding compute. Teams should choose defaults based on document characteristics, then refine after measuring retrieval quality.