

*Escribe aquí  
tu frase favorita.*

E indica aquí su autor



# Agradecimientos

Me gustaría agradecer...

También quiero destacar...

Por último...



# Índice general

Índice de figuras . . . . .	II
Índice de tablas . . . . .	III
Índice de algoritmos . . . . .	IV
1. Introducción . . . . .	1
2. Estrategias de chunkerización para LLMs . . . . .	3
2.1. Introducción . . . . .	3
2.2. Consideraciones sobre la Chunkerización . . . . .	4
2.2.1. Naturaleza del Contenido . . . . .	4
2.2.2. Modelo y Tamaño Óptimo del Chunk . . . . .	4
2.2.3. Complejidad y Longitud de las Consultas de Usuario . . . . .	4
2.2.4. Uso de los Resultados Recuperados . . . . .	4
2.2.5. Limitaciones Técnicas . . . . .	4
2.3. Estrategias de Chunkerización . . . . .	5
2.3.1. Chunkerización de Tamaño Fijo . . . . .	5
2.3.2. Chunkerización Consciente del Contenido . . . . .	5
2.3.3. Chunkerización Recursiva . . . . .	6
2.3.4. Chunkerización Especializada . . . . .	7
2.4. Conclusiones . . . . .	7
Lista de Acrónimos . . . . .	9
A. Apéndice A . . . . .	10
B. Apéndice B . . . . .	11
Bibliografía . . . . .	12



# Índice de figuras

2.1. Ejemplo de chunkerización tamaño fijo sin overload . . . . .	5
2.2. Ejemplo de chunkerización tamaño fijo con overload . . . . .	6
2.3. Ejemplo de chunkerización recursiva . . . . .	6
2.4. Ejemplo de chunkerización Especializada . . . . .	7



# Índice de tablas



# Índice de algoritmos



# Introducción

1



# Estrategias de chunkerización para LLMs

# 2

## 2.1. Introducción

La chunkerización en el ámbito de los modelos de lenguaje de gran escala (LLM) se refiere al proceso de dividir textos extensos en segmentos más manejables, conocidos como *chunks*. Esta técnica es fundamental para mejorar la gestión y el procesamiento de grandes volúmenes de información en aplicaciones de inteligencia artificial que utilizan modelos de lenguaje para entender y generar texto humano.

**Necesidad de la Chunkerización:** Los modelos de lenguaje, son inherentemente limitados por la cantidad de texto que pueden procesar en una sola instancia debido a restricciones de memoria y capacidad de cálculo. Al descomponer los textos en chunks más pequeños, se facilita que el modelo maneje datos extensos de manera eficiente, permitiendo una evaluación más rápida y precisa.

**Objetivos de la Chunkerización:** El principal objetivo de la chunkerización es maximizar la relevancia y la precisión del texto procesado. Al dividir el contenido en partes significativas y manejables, se busca preservar la coherencia semántica sin sobrecargar el modelo. Además, esta estrategia es crucial para la indexación efectiva de contenidos en bases de datos vectoriales, lo que mejora la recuperación y relevancia de los resultados en consultas específicas.

**Retos de la Chunkerización:** A pesar de sus beneficios, la chunkerización presenta varios desafíos. El más significativo es determinar el tamaño óptimo de cada chunk. Si los chunks son demasiado pequeños, pueden perder contexto necesario para una comprensión completa; si son demasiado grandes, pueden exceder las capacidades de procesamiento del modelo y diluir la relevancia del contenido. Otro reto es la selección de los puntos de corte dentro del texto, que debe hacerse de manera que se preserve la integridad del contenido semántico.

La chunkerización es un proceso esencial que requiere un cuidadoso equilibrio entre tamaño de chunk, preservación del contexto y capacidades del modelo. En las siguientes secciones, exploraremos diferentes estrategias y métodos de chunkerización, evaluando sus ventajas y limitaciones en diversos contextos de aplicación.

## 2.2. Consideraciones sobre la Chunkerización

Al desarrollar una estrategia de chunkerización eficaz, es crucial considerar varios aspectos que influirán en el rendimiento y la eficacia del modelo de lenguaje. A continuación, se detallan algunas de las consideraciones más importantes:

### 2.2.1. Naturaleza del Contenido

El tipo de contenido que se está indexando es determinante en la selección de la estrategia de chunkerización. Por ejemplo, documentos largos como artículos o libros pueden requerir un enfoque diferente en comparación con contenidos más breves como tuits o mensajes instantáneos. Esta distinción afecta tanto la elección del modelo para crear los embeddings como la estrategia de chunkerización a aplicar.

### 2.2.2. Modelo y Tamaño Óptimo del Chunk

Dependiendo del modelo utilizado para crear los embeddings, existirán tamaños de chunk en los que el modelo desempeñará mejor. Por ejemplo, algunos modelos están optimizados para trabajar con frases individuales, mientras que otros pueden manejar mejor segmentos de texto más largos. Identificar el tamaño de chunk que maximiza la calidad de los embeddings es crucial para el éxito de la estrategia de chunkerización.

### 2.2.3. Complejidad y Longitud de las Consultas de Usuario

Las expectativas sobre la longitud y complejidad de las consultas de los usuarios deben guiar la manera en que se chunkeriza el contenido. Si las consultas suelen ser cortas y específicas, es posible que se prefiera chunkerizar el contenido en segmentos más pequeños para reflejar esta especificidad. En cambio, consultas más largas y complejas podrían beneficiarse de chunks más grandes que proporcionen un contexto más amplio.

### 2.2.4. Uso de los Resultados Recuperados

El propósito final de los chunks recuperados también juega un papel crucial. Dependiendo de si los resultados serán utilizados para búsqueda semántica, respuesta a preguntas, resumen, o cualquier otro fin, la estrategia de chunkerización podría variar para optimizar la relevancia y utilidad de la información recuperada.

### 2.2.5. Limitaciones Técnicas

Las limitaciones técnicas, como el número máximo de tokens que el modelo puede procesar en una sola instancia o las restricciones de memoria, son críticas para definir el tamaño de los chunks. Estas limitaciones no solo afectan cómo se chunkeriza el contenido sino también cómo se procesa posteriormente en el modelo.

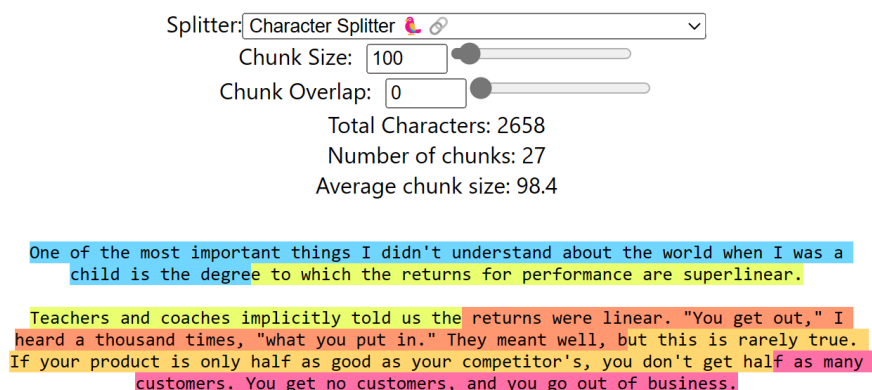
Estas consideraciones son fundamentales para desarrollar una estrategia de chunkerización que no solo sea eficiente sino también efectiva en términos de mejorar la relevancia y precisión de las respuestas del modelo.

## 2.3. Estrategias de Chunkerización

La chunkerización puede implementarse de diversas maneras, cada una con sus propios beneficios y desafíos. A continuación, exploramos algunas de las estrategias más comunes y cómo pueden optimizarse según diferentes necesidades y contextos.

### 2.3.1. Chunkerización de Tamaño Fijo

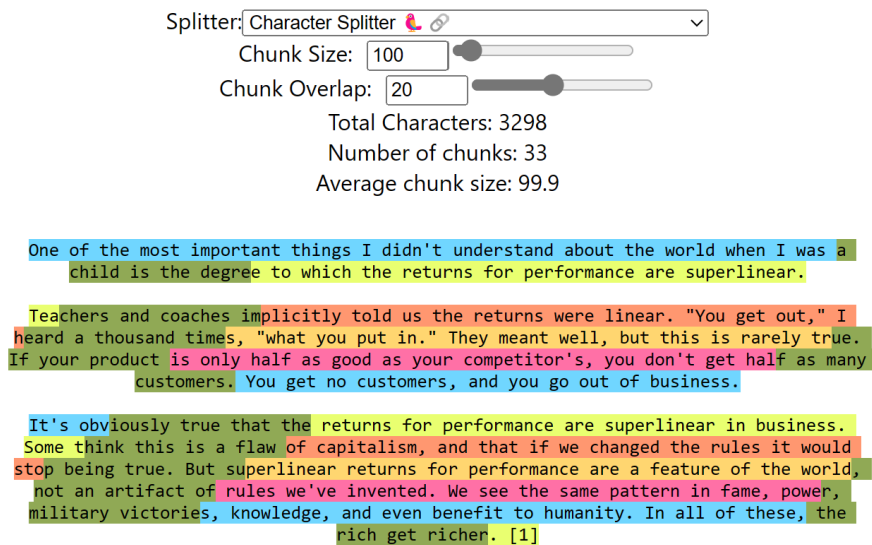
Esta estrategia consiste en dividir el texto en segmentos de un tamaño predeterminado, medido en número de tokens o caracteres. La principal ventaja de este método es su simplicidad y facilidad de implementación, ya que no requiere un análisis profundo del contenido. Sin embargo, un desafío significativo es que puede cortar frases a la mitad, perdiendo contexto o generando chunks que carecen de sentido por sí solos. Para mitigar esto, se puede optar por incluir una superposición entre los chunks, donde el final de un chunk se superpone con el inicio del siguiente, ayudando a preservar el contexto (overlap).



**Figura 2.1:** Ejemplo de chunkerización tamaño fijo sin overlap

### 2.3.2. Chunkerización Consciente del Contenido

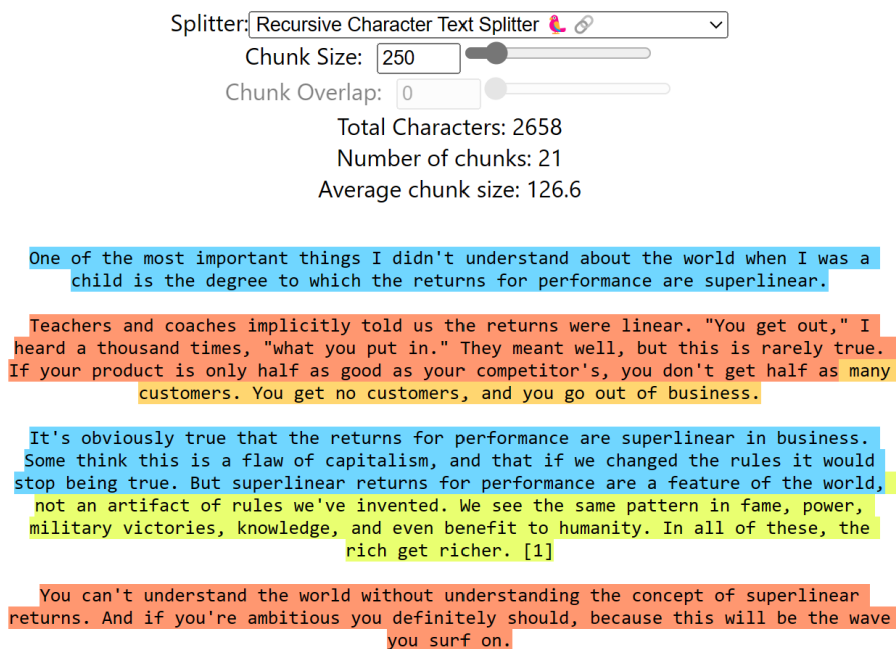
A diferencia de la chunkerización de tamaño fijo, este método tiene en cuenta el contenido textual para determinar los puntos de corte. La chunkerización consciente del contenido puede basarse en la detección de fronteras naturales en el texto, como finales de oraciones o párrafos, asegurando que cada segmento contenga una unidad completa de información. Esto ayuda a mantener la coherencia semántica de los chunks, haciendo que cada uno sea significativo por sí mismo y mejore la calidad de las incrustaciones generadas por el modelo. Este método es particularmente útil en documentos con una estructura clara y bien definida, como artículos académicos o reportes técnicos.



**Figura 2.2:** Ejemplo de chunkerización tamaño fijo con overload

### 2.3.3. Chunkerización Recursiva

Este enfoque implica un proceso iterativo y jerárquico de dividir el texto en chunks. Inicialmente, el texto se divide utilizando un criterio de separación amplio, y si los chunks resultantes aún son demasiado grandes o no cumplen con ciertos criterios, el proceso se repite en cada chunk hasta alcanzar el tamaño o la estructura deseada. La chunkerización recursiva es útil en textos largos y complejos donde los niveles múltiples de división permiten manejar mejor la diversidad y complejidad del contenido.

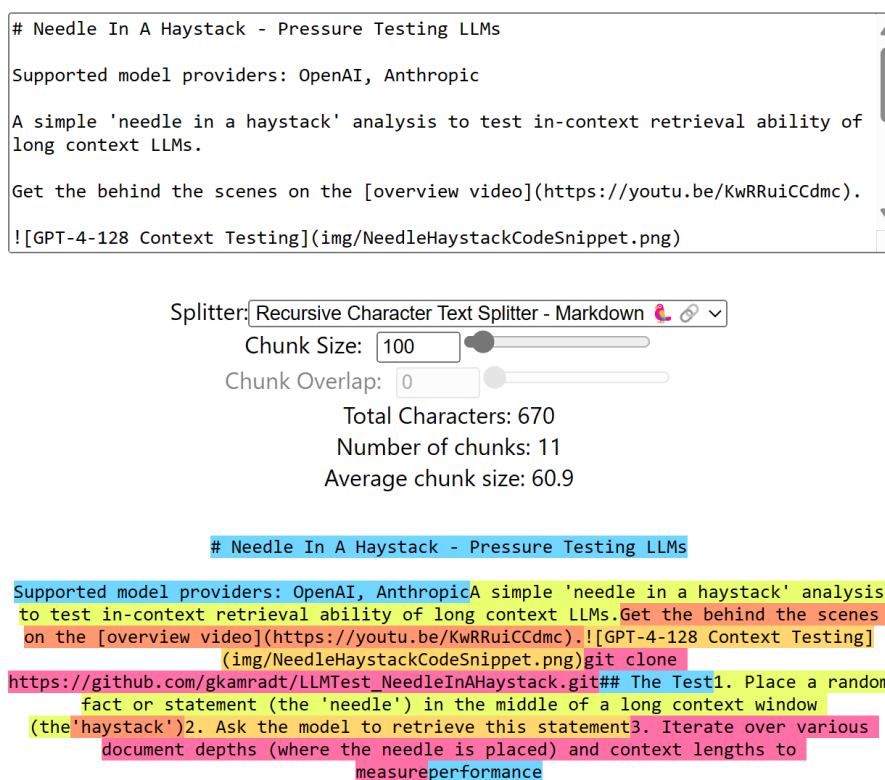


**Figura 2.3:** Ejemplo de chunkerización recursiva

### 2.3.4. Chunkerización Especializada

Esta estrategia se adapta a formatos de contenido específicos que requieren un tratamiento particular, como puede ser el caso de textos en Markdown o LaTeX. La chunkerización especializada reconoce y respeta la estructura y sintaxis propias de estos formatos, asegurando que los chunks resultantes conserven la integridad y funcionalidad del texto original. Por ejemplo, en un documento LaTeX, los chunks podrían definirse para encapsular secciones completas o subsecciones, preservando las etiquetas y comandos propios del formato.

Cada una de estas estrategias tiene sus propias fortalezas y puede ser más adecuada para diferentes tipos de textos y aplicaciones. La elección de una estrategia de chunkerización debe basarse en una evaluación cuidadosa de los requisitos del proyecto, la naturaleza del contenido y los objetivos específicos del sistema de procesamiento de lenguaje natural.



**Figura 2.4:** Ejemplo de chunkerización Especializada

### 2.3.5. Chunkerización Semántica

La chunkerización semántica es una técnica avanzada que divide el texto en unidades basadas en la relevancia semántica, utilizando modelos de creación de embeddings para evaluar la similitud entre segmentos de texto.

#### **2.3.5.1. Implementación y Ventajas**

Esta estrategia se implementa mediante la transformación de texto en representaciones vectoriales y el cálculo de su similitud, agrupando segmentos similares. Es especialmente útil para mantener la coherencia del contenido en tareas de recuperación de información y análisis de texto.

#### **2.3.5.2. Desafíos y Aplicaciones**

A pesar de su efectividad, enfrenta desafíos como la alta demanda computacional y la dificultad en determinar umbrales de similitud óptimos. Su aplicación es ideal en sistemas de búsqueda semántica y en el análisis de textos complejos en contextos académicos o de investigación.

### **2.4. Conclusiones**

La chunkerización es fundamental para mejorar la eficiencia y precisión en aplicaciones relacionadas con LLM. No existe una solución única para todos los casos, por lo que es crucial evaluar cada situación individualmente para encontrar la estrategia más efectiva.



# Lista de Acrónimos

**GAN** Red Generativa Antagónica o *Generative Adversarial Network*.

## Apéndice A



## Apéndice B

B

## Bibliografía