INSTITUTO DE PESQUISAS TECNOLÓGICAS

COMUNICAÇÃO TÉCNICA

Nº 180012

Análise de estratégias avançadas de chunking para aprimoramento de RAG

Adriana Camargo Brito
Cassia de Souza

Pôster apresentado no CONGRESSO DE MODELAGEM, SIMLAÇÃO COMPUTACIONAL E IA DO IPT, 1, 2025, São Paulo. 1 slide.

Instituto de Pesquisas Tecnológicas do Estado de São Paulo S/A - IPT
Av. Prof. Almeida Prado, 532 | Cidade Universitária ou
Caixa Postal 0141 | CEP 01064-970
São Paulo | SP | Brasil | CEP 05508-901
Tel 11 3767 4374/4000 | Fax 11 3767-4099

www.ipt.br

I CONGRESSO DE MODELAGEM, SIMULAÇÃO COMPUTACIONAL E IA DO IPT

ESTRATÉGIAS AVANÇADAS DE CHUNKING PARA APRIMORAMENTO DE RAG

Adriana Camargo de Brito[1], Cassia de Souza Carvalho[1]
[1]Instituto de Pesquisas Tecnológicas do Estado de São Paulo, Seção de Inteligência Artificial e Analytics

Introdução

Grandes Modelos de Linguagem (LLMs) possuem capacidade avançada de compreender e gerar texto, mas restringem-se ao conhecimento presente em seus dados de treinamento. Para suprir essa limitação, surgiu a técnica Retrieval-Augmented Generation (RAG), que permite incorporar informações externas como contexto. Em RAGs, os documentos são segmentados, transformados em vetores densos e indexados em bancos de dados vetoriais, possibilitando a recuperação semântica de trechos relevantes a serem fornecidos a LLMs. Com o avanço das pesquisas, surgiram estratégias que ampliam a precisão das respostas, voltadas a: segmentação de documentos, reordenação de trechos recuperados, mecanismos híbridos de busca (que combinam vetores densos e esparsos) e grafos de conhecimento.

Objetivos

Este trabalho tem como objetivo analisar o efeito de técnicas avançadas de chunking no desempenho de RAG, para identificar até que ponto diferentes granularidades de segmentação influenciam as respostas.

Metodologia / Modelagem

Foi desenvolvida uma RAG em Python com LangChain, a partir de documento PDF com oito páginas A4, sobre conectividade em cidades inteligentes. Textos, tabelas e descrições de imagens foram extraídos com PyMuPDF4LLM e segmentados com uso de quatro estratégias de chunking.

i. **Caractere Recursivo** (referência), divisão por pontuação e caracteres, com limite de tamanho fixo;
ii. **Semântico**, que cria chunks com o mesmo assunto;
iii. **SDPM** (Semantic Double-Pass Merging), que identifica mudanças de assunto pela queda de similaridade semântica entre sentenças;
iv. **LATE**, que agrupa sentenças até atingir um limite de similaridade ou tamanho do chunk.

Os parâmetros empregados nas técnicas foram: em (i), 512 caracteres de tamanho e 75 caracteres de sobreposição; em (ii, iii, iv), 512 tokens de tamanho e o modelo de vetores densos do item 2.2; em (iii, iv), 0.6 de similaridade.

Busca híbrida: vetores densos e esparsos

Os chunks foram convertidos em vetores densos e armazenados no banco de dados Qdrant, combinando vetores densos (similaridade semântica) e esparsos (palavras-chave). Os resultados são integrados por Reciprocal Rank Fusion [1], que soma os scores recíprocos para a ordenação final dos chunks. A RAG foi configurada para retornar cinco chunks com maior similaridade à pergunta do usuário.

Foram utilizados os seguintes modelos:

- paraphrase-multilingual-mpnet-base-v2 para vetores densos
- BM-25 para vetores esparsos.
- llama-3.1-8b-instant para gerar respostas
- gpt-oss-120b [9] para avaliar respostas

Mecanismos de avaliação

Foram enviadas cinco perguntas para a RAG, que teve seu desempenho avaliado com base nas seguintes métricas:

i. **Similaridade**, distância do cosseno entre a pergunta e o chunk (valor médio de todos os chunks e do primeiro chunk);
ii. **Fidelidade**, o quanto a resposta gerada é fiel ao contexto;
iii. **Relevância**, se o chunk é relevante à resposta.

As métricas (ii) e (iii) empregam um LLM para o julgamento via prompt zero-shot. As métricas (i, ii, iii) variam entre 0 e 1, com escala crescente de desempenho. Além disso, são apresentadas estatísticas dos chunks, como latência para fornecer a resposta, quantidade máxima de caracteres do chunk e tamanho médio.

Resultados e Discussão

Chunker	Num. chunks	Tamanho máximo de chunk	Tamanho médio de chunks	Latência média (s)	Similaridade		Fidelidade média	Relevância média
					Média cinco chunks	Média primeiro chunk		
Recursivo	41	510	429	2,9	0,6	0,9	0,7	0,7
SDPM	15	2219	1707	13,8	0,6	0,9	0,9	0,9
LATE	10	2381	1138	14,1	0,6	0,8	0,9	0,9
Semântico	71	1057	240	2,4	0,5	0,8	0,7	0,8

Tabela 1 – Estatísticas dos chunks e métricas de avaliação
Fonte: Dados originais da pesquisa

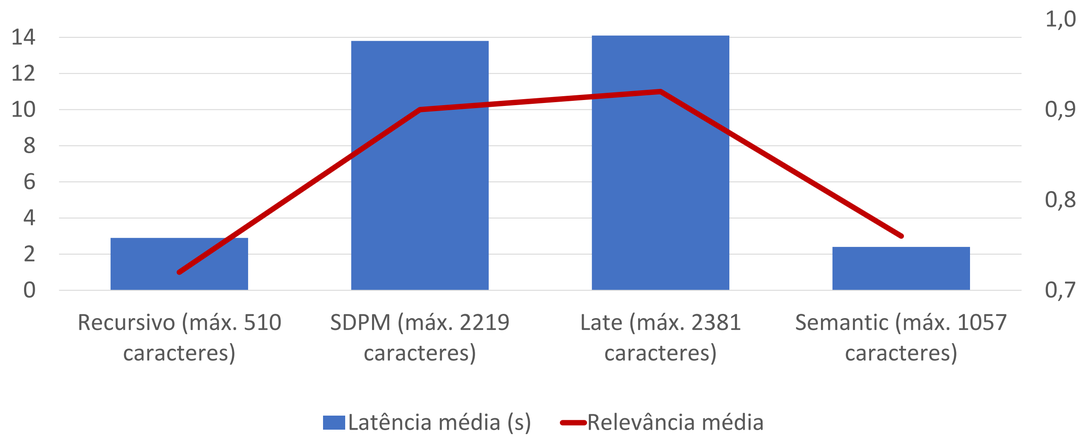

Gráfico 1 - Latência x Relevância da resposta
Fonte: Dados originais da pesquisa

Os métodos SDPM e LATE alcançaram maior fidelidade e relevância (0,9), indicando que trechos mais extensos (da ordem de 2000 caracteres) favoreceram a recuperação de contexto consistente (Tabela 1). Em contraste, com os métodos Recursivo e Semântico foram produzidos mais chunks (41 a 71), representando trechos menores (510 a 1057 caracteres), proporcionando uma latência significativamente reduzida (entre 2,4 e 2,0 s). Porém, com tais métodos, houve redução de 10 a 20% na fidelidade e relevância, possivelmente em decorrência de uma fragmentação excessiva do contexto.

A similaridade média manteve-se estável (entre 0,5 e 0,6), refletindo a diversidade semântica dos trechos recuperados..

Esses achados evidenciam um trade-off entre granularidade e desempenho: com chunking mais fino, com trechos curtos, ocorre uma otimização da velocidade de resposta, mas ocorre um comprometimento da consistência semântica (Gráfico 1). Por outro lado, com o chunking mais grosso, com trechos mais longos, há uma maior preservação do contexto, porém aumenta latência. Essa dinâmica corrobora estudos anteriores que destacam a importância do equilíbrio entre tamanho do chunk e eficiência de recuperação [2-3].

Conclusões

Este trabalho analisou o impacto de diferentes estratégias de chunking no desempenho de uma arquitetura RAG. Os resultados demonstraram que as abordagens SDPM e LATE, ao produzirem chunks mais extensos, alcançaram maiores níveis de fidelidade e relevância (0,9), indicando que a preservação de contextos amplos favorece respostas mais consistentes.

Em contrapartida, as estratégias Recursiva e Semântica, ao fragmentarem o documento em maior número de trechos menores, reduziram a latência de resposta, mas perderam de 10 a 20% em fidelidade e relevância. Há um trade-off entre granularidade e qualidade das respostas em RAGs: chunking mais fino otimiza velocidade, chunking mais grosso preserva contexto e consistência semântica.

Referências (formato reduzido, apenas as essenciais)

[1] COR M, CLARKE C, BUETTCHER S. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. Proc. SIGIR, p. 758-759, 2009.
[2] KUSH, J. et al. Introducing a new hyper-parameter for RAG: Context Window Utilization. ArXiv, 2024.
[3] MEROLA, C. et al. Reconstructing Context: Evaluating Advanced Chunking Strategies for Retrieval-Augmented Generation. ArXiv, 2025.