

Lab. 12 – Parte 2: Chunking e Avaliação de Recuperação

Aléssio Miranda Júnior

Setembro - 2026/2

Lab. 12 – Parte 2: Chunking e Avaliação de Recuperação

Capítulo em Revisão

Este conteúdo está sendo revisado. Algumas informações podem estar preliminares, conter erros de formatação ou sofrer alterações na versão final.

★ Objetivo da sessão – 40 min

Implementar alternativas de chunking e escolhê-las com consultas de desenvolvimento, reportando a recuperação em consultas held-out. Ao final, você comparará cortes fixos, recursivos e estruturais com Recall@3, MRR, precisão contextual e redundância.

Problema L2: qual fragmentação sustenta a resposta?

A DocuMind não pode escolher chunking por intuição. Usando as páginas preparadas na Parte 1, produza três índices candidatos. Use perguntas de desenvolvimento para escolher a configuração e perguntas held-out, separadas desde o início, para o relatório final. A estratégia escolhida deve equilibrar Recall@3, MRR, precisão contextual, proveniência citável, quantidade de chunks e redundância.

1 Roteiro da sessão

Tempo	Entrega
8 min	Baseline fixo orientado a tokens
10 min	Alternativas recursiva e estrutural
8 min	Indexação com proveniência preservada
10 min	Desenvolvimento, held-out e métricas
4 min	Decisão técnica; clínica extra opcional

2 1. Baseline fixo (8 min)

Um baseline simples é útil, mas corte fixo pode separar título, definição e exceção. O exemplo usa IDs de página para preservar a fonte; adapte o corte para respeitar o orçamento em tokens medido na Parte 1.

```

1 def chunks_fixos(paginas, tamanho=300, overlap=50):
2     resultado = []
3     passo = tamanho - overlap
4     if passo <= 0:
5         raise ValueError("overlap deve ser menor que tamanho")
6     for pagina in paginas:
7         tokens = tokenizer.encode(pagina["text"], add_special_tokens=
8             False)
9         for inicio in range(0, len(tokens), passo):
10            trecho = tokenizer.decode(tokens[inicio:inicio + tamanho
11                ])
12            if trecho.strip():
13                resultado.append(**pagina, "text": trecho,
14                    "chunking": "fixed-300-50")
15     return resultado

```

3 2. Alternativas (10 min)

O chunking recursivo prefere fronteiras como parágrafo e frase antes de recorrer a um corte rígido. O estrutural usa a organização do documento quando ela é confiável, por exemplo títulos extraídos de um PDF bem formado ou uma conversão prévia para Markdown.

```

1 def chunks_estruturais(secoes):
2     # secoes preservam origem e versao da extracao validada
3     # previamente.
4     return [{
5         **secao,
6         "text": f"{secao['title']}\n{secao['text']}",
7         "page_start": secao["page_range"][0],
8         "page_end": secao["page_range"][1],
9     }]

```

```

8         "section": secao["title"],
9         "chunking": "structural",
10    } for secao in secoes if secao["text"].strip()

```

Para a alternativa recursiva, você pode usar um splitter que aceite o tokenizer da Parte 1 e prefira quebra de parágrafo, quebra de linha, fim de sentença e espaço, nessa ordem. Não compare contagem de caracteres como se fosse contagem de tokens. Se a estrutura extraída for pouco confiável, declare essa limitação em vez de inventar títulos.

4 3. Indexação citável (10 min)

Indexe cada estratégia em uma coleção distinta para não misturar configurações. Todo chunk deve manter arquivo, hash, página ou intervalo de páginas, estratégia e versão do pipeline.

```

1 def indexar(collection, chunks):
2     collection.upsert(
3         ids=[f"{c['chunking']}:{i}" for i, c in enumerate(chunks)],
4         documents=[c["text"] for c in chunks],
5         metadata=[{
6             "source_file": c["source_file"],
7             "source_sha256": c["source_sha256"],
8             "page_start": c.get("page", c.get("page_start")),
9             "page_end": c.get("page", c.get("page_end")),
10            "chunking": c["chunking"],
11            "pipeline_version": c["pipeline_version"],
12        } for c in chunks],
13    )

```

Se a estratégia estrutural cobre várias páginas, armazene `page_start` e `page_end`, em vez de reduzir a proveniência a uma única página.

5 4. Desenvolvimento, held-out e métricas (10 min)

Monte perguntas cujas páginas ou seções esperadas foram verificadas por leitura humana. Separe-as antes de executar o experimento: desenvolvimento serve para escolher a configuração; held-out só é consultado após congelar essa escolha. Para cada pergunta, há sucesso se pelo menos um dos três resultados contém a evidência esperada.

```

1 desenvolvimento = [
2     {"query": "qual é a carga horária mínima?", "expected_pages":
3         {3}},
4 ]
5 held_out = [

```

```

5     {"query": "como funciona a avaliação?", "expected_pages": {7,
6         8}},
7 ]
8 def metricas_at_3(collection, casos):
9     recalls, reciprocal_ranks, contextual_precision = [], [], []
10    for caso in casos:
11        resposta = collection.query(query_texts=[caso["query"]],
12            n_results=3)
13        paginas = [m["page_start"] for m in resposta["metadatas"][0]]
14        ranks = [i + 1 for i, pagina in enumerate(paginas)
15            if pagina in caso["expected_pages"]]
16        recalls.append(bool(ranks))
17        reciprocal_ranks.append(1 / ranks[0] if ranks else 0)
18        # Marcado por leitura humana: o trecho sustenta a resposta?
19        contextual_precision.append(caso["contextual_relevant_top3"]
20            / 3)
21    return {"Recall@3": sum(recalls) / len(casos),
22        "MRR": sum(reciprocal_ranks) / len(casos),
23        "precisao_contextual": sum(contextual_precision) / len(
24            casos)}

```

Acrescente manualmente `contextual_relevant_top3` a cada caso após ler os três trechos retornados. Precisão contextual não é inferida pela página esperada: ela pergunta se o contexto sustenta de fato a resposta. Redundância pode ser a proporção de pares entre os três textos com similaridade de cosseno acima de um limiar declarado. Reporte as quatro métricas no held-out, além de latência, número de chunks e qualidade das citações.

6 5. Decisão técnica (4 min)

Compare **fixed**, **recursive** e **structural** com o mesmo modelo de embedding e Top-3. Escolha tamanho, overlap e estratégia usando desenvolvimento; congele-os e preencha a tabela com held-out. Entregue uma decisão justificada.

Estratégia	Recall@3	MRR	Prec. contextual	Redundância	Decisão/risco
Fixa					
Recursiva					
Estrutural					

◇ Informação: Clínica extra opcional: clusters exploratórios

Depois da entrega principal, aplique um agrupamento não supervisionado aos embeddings e inspecione exemplos de cada grupo. Use-o para levantar hipóteses sobre tópicos, duplicatas ou falhas de extração. Não trate grupos como categorias verdadeiras e não forneça ao algoritmo os rótulos/páginas esperadas das consultas: eles são supervisão e permanecem exclusivos da avaliação.

✓ Takeaways

Você comparou fragmentações em vez de assumir um tamanho universal. A estratégia útil é escolhida em desenvolvimento e recupera evidências citáveis no held-out, dentro do orçamento de custo e redundância aceito pelo sistema. Clusters são uma lente exploratória, não verdade de referência.