Pengantar Retrieval-Augmented Generation (RAG)

1. Pengertian RAG

Retrieval-Augmented Generation (RAG) adalah arsitektur yang menggabungkan pencarian dokumen dengan model bahasa besar. Model tidak hanya mengandalkan pengetahuan hasil pelatihan, tetapi juga menerima potongan dokumen yang relevan sebagai konteks tambahan saat menjawab pertanyaan.

Pendekatan ini diperkenalkan oleh Lewis dan kawan-kawan pada tahun 2020 di Facebook AI Research. Tujuannya adalah mengurangi halusinasi dan memungkinkan jawaban bersumber dari dokumen terbaru tanpa harus melatih ulang model dari awal.

2. Alur Kerja RAG

Sistem RAG terdiri dari dua tahap, yaitu tahap pengindeksan yang berjalan offline dan tahap penjawaban yang berjalan online. Pada tahap pengindeksan, dokumen dibaca, dipotong menjadi chunk, diubah menjadi vektor embedding, lalu disimpan dalam vector store.

Pada tahap penjawaban, pertanyaan pengguna diubah menjadi vektor dengan model embedding yang sama. Sistem lalu mengambil k chunk yang paling dekat dengan vektor pertanyaan dan menyisipkannya ke dalam prompt untuk model bahasa.

3. Embedding dan Kemiripan

Embedding adalah representasi teks berupa vektor bilangan real dengan dimensi tetap. Model MiniLM menghasilkan embedding 384 dimensi, sedangkan model berbasis BERT-base menghasilkan 768 dimensi. Teks dengan makna mirip akan menghasilkan vektor yang berdekatan.

Kemiripan antar vektor biasanya diukur dengan cosine similarity, yang bernilai antara -1 dan 1. Semakin mendekati 1, semakin searah kedua vektor tersebut. Pencarian semantik tetap dapat menemukan dokumen walaupun kata dalam pertanyaan berbeda dari teks, misalnya pertanyaan tentang kendaraan roda empat yang cocok dengan dokumen tentang mobil.

4. Chunking

Chunk yang terlalu kecil kehilangan konteks, sedangkan chunk yang terlalu besar mencampur beberapa topik sehingga skor kemiripannya menjadi encer. Praktik yang umum adalah ukuran 500 sampai 1000 karakter dengan overlap 10 sampai 20 persen.

Overlap mencegah informasi penting terpotong tepat di batas antar chunk. Pemotongan berbasis kalimat atau paragraf lebih rapi daripada pemotongan karakter murni, tetapi ukuran chunk menjadi tidak seragam. Tabel perlu diperlakukan khusus karena baris tabel kehilangan makna bila terpisah dari baris judul kolomnya.

Tabel 1. Perbandingan ukuran chunk

Ukuran	Kelebihan	Kekurangan	Cocok untuk
500 karakter	Presisi tinggi, hemat token	Konteks sempit, kalimat sering terpotong	FAQ dan pertanyaan faktual singkat
1000 karakter	Seimbang antara konteks dan presisi	Kadang memuat dua topik	Dokumen laporan umum

1500 karakter	Konteks luas	Skor encer dan boros token	Dokumen naratif panjang

Sumber: ringkasan penulis.

5. Vector Store

Vector store menyimpan embedding dan mendukung pencarian tetangga terdekat. FAISS adalah pustaka buatan Meta untuk pencarian tetangga terdekat yang sangat cepat. ChromaDB menyimpan embedding bersama metadata seperti nama file dan nomor halaman, sedangkan Qdrant menyediakan layanan server tersendiri.

Untuk koleksi besar, banyak vector store memakai indeks HNSW yang mencari tetangga terdekat secara aproksimasi demi kecepatan, dengan sedikit pengorbanan akurasi.

6. Evaluasi Retrieval

Hit@k adalah proporsi pertanyaan uji yang jawaban benarnya muncul di antara k chunk teratas. Sebagai contoh, bila 12 dari 15 pertanyaan terjawab pada lima chunk teratas, maka Hit@5 bernilai 0,8. Metrik lain yang sering dipakai adalah MRR dan Recall@k.

Disarankan menyiapkan minimal 15 sampai 30 pertanyaan uji lengkap dengan kata kunci jawaban. Bandingkan beberapa pengaturan chunk pada pertanyaan yang sama agar perbedaannya adil.

7. Halusinasi dan Sitasi

Halusinasi adalah keluaran model yang terdengar meyakinkan tetapi tidak didukung fakta. RAG mengurangi halusinasi, tetapi tidak menghilangkannya sepenuhnya, terutama bila chunk yang diambil tidak relevan.

Sistem yang baik menampilkan sumber berupa nama file dan nomor halaman sehingga pengguna dapat memeriksa jawaban. Bila chunk teratas tidak relevan, model sebaiknya menjawab bahwa informasi tidak ditemukan daripada mengarang.