

RAG

Enzo Bogнар

Por que usar RAG (Retrieval-Augmented Generation)?

- LLMs possuem conhecimento limitado ao que foi treinado.
- Sem acesso externo, modelos podem alucinar informações.
- Muitos problemas reais dependem de documentos, relatórios, PDFs, textos.
- Precisamos de um método para consultar informações externas com precisão.

Sistema RAG implementado em Python + HuggingFace

- **Tecnologias utilizadas:**

SentenceTransformers / HuggingFace → gerar embeddings

FAISS → busca vetorial

FLAN-T5 (HuggingFace Transformers) → gerar resposta

Google Colab → ambiente de execução

- **Funcionamento do sistema:**

Usuário faz uma pergunta.

Pergunta é convertida em embedding.

FAISS busca os documentos mais semelhantes (similaridade vetorial).

Esses documentos são enviados ao modelo generativo.

O modelo responde usando as informações recuperadas.

- **Resultado esperado:**

O modelo responde com base em fatos da base de conhecimento, evitando alucinações.

Demonstração e Resultados

Conclusão:

O sistema funciona como um pipeline RAG básico, mostrando recuperação + geração em funcionamento real.

```
[9]
✓ 18s
query = "Para que serve o RAG?"
answer, retrieved = rag_answer(query)

print("🔥 Pergunta:", query)
print("📄 Documentos Recuperados:\n", retrieved)
print("🗨️ Resposta do Modelo:\n", answer)
```

*** Both `max_new_tokens` (=256) and `max_length` (=150) seem to have been set. `max_new_tokens` will take precedence. Please refer to the documentation for more information. (https://huggingface.co/docs/transformers/main/en/main_classes/text_generation)

🔥 Pergunta: Para que serve o RAG?

📄 Documentos Recuperados:
LLMs podem usar RAG para acessar informações externas sem precisar memorizar tudo nos pesos do modelo.
FAISS é uma biblioteca do Facebook para busca vetorial em grandes coleções de embeddings.

🗨️ Resposta do Modelo:
LLMs can use RAG to access external information without precisar memorizar all nos pesos do model. FAISS is a biblioteca do Facebook para busca vetorial em grande coleções de embeddings.