RAG (Retrieval-Augmented Generation)

RAG (česky: generování rozšířené o vyhledávání) je technika v oblasti umělé inteligence, která umožňuje velkým jazykovým modelům přistupovat k externím datům v reálném čase, aniž by bylo nutné model znovu trénovat.

Zatímco standardní model odpovídá pouze na základě svých „vnitřních znalostí“ (které mají datum uzávěrky), RAG mu umožňuje nejdříve vyhledat relevantní informace v soukromých dokumentech nebo na internetu a až poté sestavit odpověď.

Proč používat RAG?

- **Aktuálnost:** Model může pracovat s informacemi starými jen pár sekund (např. aktuální zprávy nebo stav skladu).
- **Snížení halucinací:** Model je nucen podložit svou odpověď nalezenými fakty. Pokud informaci nenajde, může říct „nevím“, místo aby si vymýšlel.
- **Soukromí a bezpečnost:** Umožňuje bezpečně pracovat s firemními daty (např. v DokuWiki), která nikdy neopustí vaši infrastrukturu a nejsou součástí veřejného tréninku modelu.
- **Citace zdrojů:** RAG systémy mohou přesně ukázat, ze kterého dokumentu informaci čerpaly.

Jak RAG funguje (Architektura)

Proces probíhá v několika krocích, které se spustí po zadání dotazu uživatelem:

1. Vektorizace (Indexing)

Předem připravené dokumenty se rozdělí na menší kusy (chunks) a převedou se na číselné vektory (embeddings) pomocí speciálního modelu. Tyto vektory se uloží do **vektorové databáze**.

2. Vyhledávání (Retrieval)

Když položíte dotaz, systém jej také převede na vektor a najde v databázi sémanticky nejpodobnější kusy textu.

3. Rozšíření (Augmentation)

Původní dotaz se „obalí“ nalezenými texty. Vznikne nový, rozšířený prompt: „`Zde jsou informace z naší dokumentace: [Nalezený text]. Na základě těchto informací odpověz na otázku: [Původní dotaz].`“

4. Generování (Generation)

LLM zpracuje tento obří prompt a vygeneruje odpověď, která je přesná a podložená fakty.

Srovnání: RAG vs. Fine-tuning

Vlastnost	RAG	Fine-tuning
Znalost nových dat	Okamžitá (stačí přidat soubor)	Vyžaduje nový trénink (dny/týdny)
Přesnost (fakta)	Velmi vysoká	Střední (může stále halucinovat)
Náklady	Nízké (provoz databáze)	Vysoké (výpočetní výkon GPU)
Transparentnost	Vysoká (uvádí zdroje)	Nízká (černá skříňka)

Klíčové technologie pro RAG

- **Vektorové databáze:** Pinecone, Milvus, Weaviate, nebo rozšíření **pgvector** pro PostgreSQL.
- **Frameworky: LangChain** nebo **LlamaIndex** (nástroje pro „pospojování“ databáze, modelu a logiky).
- **Embedding modely:** Modely od OpenAI, Cohere nebo open-source modely z Hugging Face (např. BERT varianty).

Výzvy při implementaci

- **Kvalita dat:** Pokud jsou zdrojové dokumenty chaotické, bude špatná i odpověď (tzv. „Garbage In, Garbage Out“).
- **Strategie dělení (Chunking):** Jak velké kusy textu ukládat, aby neztratily kontext, ale zároveň nebyly příliš dlouhé.
- **Reranking:** Dodatečné seřazení výsledků vyhledávání pro výběr těch skutečně nejlepších.

> **Příklad z praxe:** Firemní chatbot pro zaměstnance. Místo aby se ptali kolegů, položí dotaz chatbotu, který pomocí RAG prohledá firemní PDF směrnice, Wiki stránky a Slack historii a okamžitě odpoví s odkazem na zdroj.

Zpět na AI rozcestník