Engineer

Retrieval-augmented generation

📖 att generera svar utifrån dokument hämtade för frågan

Vad det är

Retrieval-augmented generation är ett sätt att besvara en fråga med en språkmodell i två steg. Först söker applikationen i en samling dokument efter de utdrag som mest sannolikt innehåller svaret. Sedan lägger den in utdragen i modellens indata bredvid frågan, och modellen skriver svaret utifrån dem i stället för utifrån det den tog till sig under träningen. Namnet, som oftast förkortas RAG, kommer från en forskningsartikel från 2020 om kunskapsintensiva språkuppgifter som att besvara öppna frågor.

Artikeln beskrev två slags minne. En förtränad modell bevarar det den har lärt sig i sina parametrar, och det kallade författarna parametriskt minne. Ett separat index över dokument, som en tränad sökmodell, en så kallad retriever, söker i, är icke-parametriskt minne. Deras system kopplade en förtränad sequence-to-sequence-modell till ett tätt vektorindex över Wikipedia, och de jämförde två sätt att använda träffarna: det ena låter hela svaret bygga på samma utdrag, det andra kan hämta från olika utdrag för varje token det skriver. På tre referenstester för frågebesvarande i öppna domäner satte systemet nya rekord, och författarna fann att texten det genererade var mer specifik, mer varierad och mer faktamässigt korrekt än texten från en jämförbar modell utan sökning.

En applikation kan bygga samma mönster av delar den redan har: ett sökindex över de egna dokumenten och en stor språkmodell som den anropar via ett API. Sökdelen är vanlig sökning, och båda familjerna av sökning dyker upp här. Sparse retrieval matchar ord, på det sätt som fulltextsökning rangordnar dokument med en poängfunktion som BM25. Dense retrieval gör om frågan och varje utdrag till vektorer av tal och returnerar de utdrag vars vektorer ligger närmast, och det är vektorsökning. En artikel från 2020 om dense passage retrieval fann att en tät retriever, tränad på ett måttligt antal par av frågor och utdrag, slog en stark

BM25-baslinje med 9 till 19 procentenheter, mätt på hur ofta rätt utdrag fanns bland de tjugo första resultaten. Ett system kan köra båda och slå ihop de två listorna.

Använd begreppet i ett beslut

Använd ”Vad är retrieval-augmented generation?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Dina data samlade där AI-agenter kan nå dem, med en enda version av varje kund, utan att någon ser mer än de får, och med varje svar spårat tillbaka till källan.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Gjorde pipelinen för geodata 50 gånger snabbare — 50x snabbare datapipelines för geodata

Gjorde pipelinen för geodata 50 gånger snabbare Hör av er →

Varför det spelar roll

En språkmodell känner bara till det som stod i träningstexten, och den texten slutar vid ett fast datum och har aldrig innehållit dina avtal, dina prislistor eller anteckningarna från förra veckans möte. RAG-artikeln pekar på två svagheter hos kunskap som ligger i parametrar: det är svårt att visa var ett svar kom ifrån, och svårt att uppdatera kunskapen. Sökningen tar hand om båda. Svaret kan hänvisa till de utdrag modellen fick, så att en läsare kan öppna källan och kontrollera den, och ny kunskap når modellen i samma ögonblick som ett dokument kommer in i indexet, helt utan en träningskörning.

Det placerar också frågan om vem som får se vad där den hör hemma. En sökning kan köras med behörigheterna hos den som frågar, så att modellen bara får se utdrag som den personen själv hade kunnat öppna. I en modell som hade tränats på samma dokument finns inte den gränsen: allt den har lärt sig är tillgängligt för alla som använder den.

Priset är att svaret aldrig blir bättre än det sökningen returnerar. En modell som får fel utdrag skriver ett flytande svar utifrån dem, och en modell som inte får något användbart svarar kanske ur minnet ändå. Det mesta av arbetet i ett system byggt på retrieval-augmented generation ligger därför i datan och sökningen, inte i modellen.

Så ser det ut i praktiken

Spåret nedan följer en fråga genom ett sådant system. Sökningen körs med behörigheterna hos användaren som frågar, varje utdrag kommer med filen och sidan det kommer från, och instruktionerna ber modellen svara utifrån utdragen och ingenting annat:

```
fråga:     Vilka leveransvillkor gäller för order 4471?
användare: sälj-3 (roll: sälj, region: norr)

träffar, filtrerade till det sälj-3 får läsa:
  [1] avtal/ramavtal-2025.pdf, sida 4
      ”Varor levereras till köparens lager,
       om inte ordern anger något annat.”
  [2] order/4471.txt, rad 12
      ”Leveransvillkor: se ramavtalet.”

instruktioner:
  Svara bara utifrån utdragen nedan och hänvisa till dem som [n].
  Om de inte besvarar frågan, säg det.

svar:
  Order 4471 hänvisar till ramavtalet [2], som säger att
  varor levereras till köparens lager [1].
```

Varje del av spåret är data som systemet kan kontrollera. Ett test kan bekräfta att rätt utdrag kom tillbaka för en känd fråga, ett skript kan bekräfta att varje `[n]` i svaret hänvisar till ett utdrag som faktiskt hittades, och en person som läser svaret kan slå upp sidan 4 i avtalet.

Förarbetet som detta kräver är ämnet för sidan om företagsdata redo för AI-agenter här på webbplatsen. Den beskriver hur gamla PDF:er, inskannade dokument och e-postarkiv görs om till sökbar text med källan sparad bredvid, och hur varje rad bär med sig varifrån den kom och när, alltså systemet, inläsningen, filen och sidan, så att en agent kan hänvisa till posten bakom en siffra och en människa kan öppna den. Sidan lägger också reglerna i datan och inte i prompten: agenten agerar med sin användares behörigheter, aldrig en administratörs, och varje läsning loggas med personen den gjordes för.

Var det går fel

Sökningen missar, och modellen svarar ändå. När rätt utdrag inte hittas får modellen ändå något och skriver ändå ett svar. Mät sökningen för sig: skriv ner riktiga frågor tillsammans med de utdrag som besvarar dem, och kontrollera hur ofta de utdragen finns bland resultaten modellen får. Be modellen säga till när utdragen inte besvarar frågan, och ta med sådana frågor bland testfallen.

Utdrag som skär svaret i två. När dokument delas upp i bitar av fast längd skiljs en tabell från sin rubrik och en klausul från sitt undantag, och varje halva vilseleder när den hittas ensam. Dela längs den struktur som dokumentet redan har, alltså styckena, sidorna och tabellraderna, och håll dokumentets titel och placering ihop med varje bit.

Indexet bryr sig inte om vem som frågar. Ett index som har byggts av en tjänst som kan läsa allt returnerar utdrag ur dokument som den som frågar inte får öppna, och modellen sammanfattar dem för den personen. Filtrera på användarens behörigheter inne i sökningen, i indexet eller databasen, innan något når modellen. En regel i prompten är inte åtkomstkontroll.

Instruktioner planterade i ett dokument. Text som hittas är text som någon annan har skrivit, och modellen kan inte på ett tillförlitligt sätt skilja ett utdrag från en instruktion. NIST:s profil för generativ AI beskriver indirekt prompt injection som attacker mot applikationer byggda på språkmodeller genom data som sannolikt kommer att hämtas, och en artikel från 2023 demonstrerade sådana attacker mot verkliga system. Behandla varje utdrag som hittas som indata du inte kan lita på, låt inget utdrag utlösa en handling utan en kontroll i kod, och ge verktygen som modellen kan anropa inte fler behörigheter än användaren själv har.

Ett index som släpar efter. Ett dokument ändras eller raderas, och indexet innehåller fortfarande de gamla utdragen, så systemet svarar självsäkert utifrån ett avtal som inte längre gäller. Uppdatera indexet varje gång en källa ändras, ta bort utdrag när deras dokument raderas, och visa datumet för varje källa bredvid svaret.

Källhänvisningar som inte säger det svaret påstår. En modell kan hänvisa till utdrag [2] för en mening som utdrag [2] inte stöder, och NIST påpekar att genererade källhänvisningar själva kan vara konfabulerade. Kontrollera i kod att

varje hänvisning pekar på ett utdrag som hittades, och låt människor stickprovsgranska svar mot deras källor som en fast del av testarbetet.

Läs mer

- Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks — artikeln från 2020 som introducerade RAG-modellerna och jämförde de två varianterna på frågebesvarande och textgenerering.
- Dense Passage Retrieval for Open-Domain Question Answering — en tät retriever, tränad på par av frågor och utdrag och mätt mot BM25 på att hitta det utdrag som innehåller svaret.
- NIST AI 600-1: Generative Artificial Intelligence Profile — NIST:s riskprofil för generativ AI, med konfabulering, skydd av personuppgifter och indirekt prompt injection.
- Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection — hur instruktioner gömda i hämtade data tar över en applikation, med attacker mot verkliga system.

Ordlista

Välj utifrån era faktiska förutsättningar

Ta med ett representativt dataset, frågorna det ska besvara och dagens flaskhals. Vi kan avtala mått på kvalitet, genomströmning och återställning.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är retrieval-augmented generation?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

welcome@engineer.company +45 50 10 61 84 Köpenhamn, Danmark

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/retrieval-augmented-generation/