Engineer

Retrieval-augmented generation

📖 å generere svar ut fra dokumenter hentet til spørsmålet

Hva det er

Retrieval-augmented generation er en måte å besvare et spørsmål med en språkmodell på i to steg. Først søker applikasjonen i en samling dokumenter etter de utdragene som mest sannsynlig inneholder svaret. Deretter legger den utdragene inn i modellens inndata ved siden av spørsmålet, og modellen skriver svaret ut fra dem i stedet for ut fra det den tok opp i seg under treningen. Navnet, som som regel forkortes til RAG, kommer fra en forskningsartikkel fra 2020 om kunnskapskrevende språkoppgaver som å besvare åpne spørsmål.

Artikkelen beskrev to slags hukommelse. En forhåndstrent modell tar vare på det den har lært, i parameterne sine, og det kalte forfatterne parametrisk hukommelse. En separat indeks over dokumenter, som en trent søkemodell, en såkalt retriever, søker i, er ikke-parametrisk hukommelse. Systemet deres koblet en forhåndstrent sequence-to-sequence-modell til en tett vektorindeks over Wikipedia, og de sammenlignet to måter å bruke funnene på: den ene lar hele svaret bygge på de samme utdragene, den andre kan hente fra ulike utdrag for hvert token den skriver. På tre referansetester for besvarelse av spørsmål i åpne domener satte systemet nye rekorder, og forfatterne fant at teksten det genererte, var mer spesifikk, mer variert og mer faktisk korrekt enn teksten fra en sammenlignbar modell uten søk.

En applikasjon kan bygge det samme mønsteret av deler den allerede har: en søkeindeks over sine egne dokumenter og en stor språkmodell den kaller via et API. Søkedelen er vanlig søk, og begge familiene av søk dukker opp her. Sparse retrieval matcher ord, slik fulltekstsøk rangerer dokumenter med en poengfunksjon som BM25. Dense retrieval gjør spørsmålet og hvert utdrag om til vektorer av tall og returnerer utdragene der vektorene ligger nærmest, og det er vektorsøk. En artikkel fra 2020 om dense passage retrieval fant at en tett retriever, trent på et beskjedent antall par av spørsmål og utdrag, slo en sterk

BM25-baseline med 9 til 19 prosentpoeng, målt på hvor ofte det riktige utdraget var blant de tjue første resultatene. Et system kan kjøre begge og flette de to listene.

Bruk begrepet i en beslutning

Bruk «Hva er retrieval-augmented generation?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Dataene dine samlet der AI-agenter kan nå dem, med én versjon av hver kunde, uten at noen får se mer enn de skal, og med hvert svar sporet tilbake til kilden.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Gjorde en geodatapipeline 50 ganger raskere — 50x raskere pipelines for geografiske data

Gjorde en geodatapipeline 50 ganger raskere Ta kontakt →

Hvorfor det er viktig

En språkmodell kjenner bare det som sto i treningsteksten, og den teksten slutter på en fast dato og har aldri inneholdt kontraktene dine, prislistene dine eller referatet fra forrige ukes møte. RAG-artikkelen peker på to svakheter ved kunnskap som ligger i parametere: det er vanskelig å vise hvor et svar kom fra, og vanskelig å oppdatere kunnskapen. Søket tar seg av begge deler. Svaret kan vise til utdragene modellen fikk, så en leser kan åpne kilden og kontrollere den, og ny kunnskap når fram til modellen i det øyeblikket et dokument kommer inn i indeksen, helt uten en treningskjøring.

Det plasserer også spørsmålet om hvem som får se hva, der det hører hjemme. Et søk kan kjøre med rettighetene til den som spør, slik at modellen bare får se utdrag som den personen selv kunne ha åpnet. I en modell som var trent på de samme dokumentene, finnes ikke den grensen: alt den har lært, er tilgjengelig for alle som bruker den.

Prisen er at svaret aldri blir bedre enn det søket returnerer. En modell som får feil utdrag, skriver et flytende svar ut fra dem, og en modell som ikke får noe brukbart, svarer kanskje ut fra hukommelsen likevel. Mesteparten av arbeidet i et system bygd på retrieval-augmented generation ligger derfor i dataene og søket, ikke i modellen.

Slik ser det ut i praksis

Sporet nedenfor følger ett spørsmål gjennom et slikt system. Søket kjører med rettighetene til brukeren som spør, hvert utdrag kommer med filen og siden det stammer fra, og instruksjonene ber modellen svare ut fra utdragene og ingenting annet:

```
spørsmål: Hvilke leveringsbetingelser gjelder for ordre 4471?
bruker:   salg-3 (rolle: salg, region: nord)

funnet, filtrert til det salg-3 får lese:
  [1] kontrakter/rammeavtale-2025.pdf, side 4
      «Varer leveres til kjøperens lager,
       med mindre ordren angir noe annet.»
  [2] ordrer/4471.txt, linje 12
      «Leveringsbetingelser: jf. rammeavtalen.»

instruksjoner:
  Svar bare ut fra utdragene nedenfor, og vis til dem som [n].
  Hvis de ikke besvarer spørsmålet, så si det.

svar:
  Ordre 4471 viser til rammeavtalen [2], som sier at
  varer leveres til kjøperens lager [1].
```

Hver del av sporet er data systemet kan kontrollere. En test kan bekrefte at de riktige utdragene kom tilbake for et kjent spørsmål, et skript kan bekrefte at hver `[n]` i svaret viser til et utdrag som faktisk ble funnet, og en person som leser svaret, kan slå opp på side 4 i kontrakten.

Forarbeidet dette krever, er temaet for siden om bedriftsdata klare for AI-agenter her på nettstedet. Den beskriver hvordan gamle PDF-er, skanninger og e-postarkiver gjøres om til søkbar tekst med kilden lagret ved siden av, og hvordan hver rad tar vare på hvor og når den kom inn, nemlig systemet, innlastingen, filen og siden, slik at en agent kan vise til posten bak et tall og et menneske kan åpne den. Siden legger også reglene i dataene og ikke i prompten: agenten handler med brukerens rettigheter, aldri med en administrators, og hvert oppslag logges med hvem det ble gjort for.

Hvor det går galt

Søket bommer, og modellen svarer likevel. Når det riktige utdraget ikke blir funnet, får modellen fortsatt noe og skriver fortsatt et svar. Mål søket for seg: skriv ned ekte spørsmål sammen med utdragene som besvarer dem, og kontroller hvor ofte de utdragene er blant resultatene modellen får. Be modellen si fra når utdragene ikke besvarer spørsmålet, og ta med slike spørsmål blant testsakene.

Utdrag som skjærer svaret i to. Når dokumenter deles i biter av fast lengde, blir en tabell skilt fra overskriften sin og en klausul fra unntaket sitt, og hver halvdel villeder når den blir funnet alene. Del langs strukturen dokumentet allerede har, altså avsnittene, sidene og tabellradene, og hold dokumentets tittel og plassering sammen med hver bit.

Indeksen bryr seg ikke om hvem som spør. En indeks som er bygd av en tjeneste som kan lese alt, returnerer utdrag fra dokumenter den som spør, ikke får åpne, og modellen oppsummerer dem for vedkommende. Filtrer på brukerens rettigheter inne i søket, i indeksen eller databasen, før noe når modellen. En regel i prompten er ikke tilgangskontroll.

Instruksjoner plantet i et dokument. Funnet tekst er tekst noen andre har skrevet, og modellen kan ikke pålitelig skille et utdrag fra en instruksjon. NISTs profil for generativ AI beskriver indirekte prompt injection som angrep på applikasjoner bygd på språkmodeller gjennom data som sannsynligvis blir hentet fram, og en artikkel fra 2023 demonstrerte slike angrep mot ekte systemer. Behandle hvert funnet utdrag som inndata du ikke kan stole på, la ingen utdrag utløse en handling uten en kontroll i kode, og gi verktøyene modellen kan kalle, ikke flere rettigheter enn brukeren selv har.

En indeks som henger etter. Et dokument blir endret eller slettet, og indeksen inneholder fortsatt de gamle utdragene, så systemet svarer selvsikkert ut fra en kontrakt som ikke lenger gjelder. Oppdater indeksen hver gang en kilde endres, fjern utdrag når dokumentet deres blir slettet, og vis datoen for hver kilde ved siden av svaret.

Kildehenvisninger som ikke sier det svaret påstår. En modell kan vise til utdrag [2] for en setning som utdrag [2] ikke underbygger, og NIST påpeker at genererte kildehenvisninger selv kan være konfabulerte. Kontroller i kode at hver

henvisning peker på et utdrag som ble funnet, og la mennesker ta stikkprøver av svar mot kildene deres som en fast del av testarbeidet.

Les mer

- Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks — artikkelen fra 2020 som introduserte RAG-modellene og sammenlignet de to variantene på besvarelse av spørsmål og generering av tekst.
- Dense Passage Retrieval for Open-Domain Question Answering — en tett retriever, trent på par av spørsmål og utdrag og målt mot BM25 på å finne utdraget som inneholder svaret.
- NIST AI 600-1: Generative Artificial Intelligence Profile — NISTs risikoprofil for generativ AI, med konfabulering, personvern og indirekte prompt injection.
- Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection — hvordan instruksjoner gjemt i funnede data tar over en applikasjon, med angrep på ekte systemer.

Ordliste

Velg ut fra deres faktiske rammer

Ta med et representativt datasett, spørsmålene det skal besvare og dagens flaskehals. Vi kan avtale målinger av kvalitet, gjennomstrømning og gjenoppretting.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er retrieval-augmented generation?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

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

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/retrieval-augmented-generation/