Engineer

Stor språkmodell

et nevralt nettverk som forutsier neste token i en tekst

Hva det er

En stor språkmodell er et nevralt nettverk som er trent på en svært stor mengde tekst til å gjøre én ting: ut fra teksten så langt å gi hvert token som kan komme som det neste, en sannsynlighet. Et token er en kort tekstbit, ofte et helt ord, noen ganger en del av et ord eller et enkelt tegn. Å skrive et svar er det samme steget gjentatt i en løkke. Programmet velger et token ut fra sannsynlighetene, legger det til bakerst i teksten og spør på nytt, helt til modellen gir et token som betyr stopp, eller en lengdegrense er nådd. NISTs profil for generativ AI, utgitt i juli 2024, sier det på én linje: store språkmodeller forutsier neste token eller ord i en setning. Forkortelsen LLM, etter engelsk large language model, betyr det samme, og det er den formen profilen selv bruker på engelsk.

Arkitekturen bak de siste årenes modeller heter Transformer og ble beskrevet i en artikkel om maskinoversettelse fra 2017. Den kvittet seg med de rekurrente og konvolusjonelle lagene som tidligere modeller var bygd av, og bygde utelukkende på attention, en mekanisme som lar hver posisjon i teksten se direkte på alle de andre. Det gjorde treningen langt enklere å fordele over mange prosessorer, og artikkelens modell ble trent på vesentlig kortere tid enn forgjengerne.

«Stor» viser til antallet parametere, vektene nettverket lærer under treningen. En artikkel fra 2020 beskrev en modell med 175 milliarder av dem, ti ganger så mange som noen tidligere ikke-glissen språkmodell, og viste at en modell av den størrelsen kunne ta fatt på en ny oppgave ut fra en håndfull gjennomarbeidede eksempler i inndataene sine, helt uten videre trening.

En modell som bare er trent til å fortsette tekst, gjør nettopp det: den fortsetter teksten, og det er ikke det samme som å gjøre det den ble bedt om. En artikkel fra 2022 beskrev steget som gjør en slik modell til en assistent. Mennesker skrev

svarene de ønsket seg på et sett forespørsler, og modellen ble finjustert på dem. Deretter rangerte mennesker modellens egne svar, og den ble trent videre på rangeringene, en metode som kalles reinforcement learning from human feedback. I artikkelens evalueringer foretrakk folk svarene fra en modell med 1,3 milliarder parametere, trent på denne måten, framfor svarene fra den opprinnelige modellen med hundre ganger så mange parametere.

Bruk begrepet i en beslutning

Bruk «Hva er en stor språkmodell?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. En agent koblet til systemene du allerede bruker, testet på saker dine egne fagfolk skriver, under oppsyn av folkene dine og aldri låst til én modell.

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. Bygde repoets kontrollmotor med 268 kontroller — 268 commit-kontroller, og 146 tester av selve kontrollene

Bygde repoets kontrollmotor med 268 kontroller Ta kontakt →

Hvorfor det er viktig

En språkmodell kan ta seg av arbeid som er vanskelig å dekke med faste regler: å legge en e-post skrevet i fritekst i riktig mappe, å hente feltene ut av en skannet følgeseddel, å skrive et utkast til svar på kundens eget språk, å oppsummere en lang tråd. Arbeid som før krevde at et menneske leste hver eneste sak, kan nå lages som utkast av programvare og kontrolleres av et menneske.

Den samme egenskapen som gjør modellen nyttig, gjør den også vanskelig å bygge på. Utdataene trekkes som regel tilfeldig fra en sannsynlighetsfordeling, så samme inndata kan gi ulike svar fra kjøring til kjøring, og modellen formulerer seg like flytende enten den har rett eller ikke. NIST kaller selvsikkert framført, feil utdata for konfabulering, bedre kjent som hallusinasjon, og beskriver det som en naturlig følge av hvordan modellene er bygd: de lager tekst som passer til statistikken i teksten de ble trent på, og den er ofte riktig og noen ganger ikke. NIST påpeker også at en modell kan konfabulere begrunnelsen eller kildehenvisningene som ser ut til å underbygge et svar. Et feil svar ser til forveksling ut som et riktig, så den eneste måten å vite hvor ofte en modell har rett i ditt arbeid på, er å måle den på ditt arbeid.

Det finnes heller ikke én modell å binde seg til. Nye modeller kommer jevnlig, to av dem kan oppføre seg ganske ulikt på samme inndata, og det er leverandøren av modellen en applikasjon kaller, som bestemmer når den endrer seg. Et team

som bygger på en språkmodell, bygger på en del det ikke selv styrer, og arbeidet som varer, er alt rundt modellen: sakene som måler den, verktøyene den får kalle, og reglene for hva den får gjøre på egen hånd.

NISTs profil nevner tolv risikoområder for generativ AI, blant dem konfabulering, personvern, informasjonssikkerhet og immaterialrett. Den er skrevet for organisasjoner som styrer risiko, mer enn for utviklere, men den fungerer godt som sjekkliste for et team som skal avgjøre hva en funksjon bygd på en modell får røre.

Slik ser det ut i praksis

En funksjon bygd på en språkmodell begynner med nedskrevne saker, ikke med en prompt. Hver sak består av ekte inndata og svaret en person som kan arbeidet, ville godkjent, og sakene ligger i en fil under versjonskontroll ved siden av koden:

```
# cases/delivery-notes.yaml
- id: note-0142
  input: scans/note-0142.txt
  expect:
    supplier: "Example Freight"
    reference: "DN-58213"
    pallets: 6
- id: note-0157
  input: scans/note-0157.txt
  expect:
    supplier: "Sample Haulage"
    reference: "DN-59002"
    pallets: 14
```

Hvilken modell som tar hvilken type oppgave, er en innstilling, så en liten og rask modell kan stå for uttrekket mens en større skriver utkast til svarene som krever en vurdering:

```
# routing.yaml
jobs:
  extract-delivery-note:
    model: small-fast
    cases: cases/delivery-notes.yaml
```

```
  draft-claim-reply:
    model: large
    cases: cases/claim-replies.yaml
```

Hver endring kjøres mot alle sakene før den tas i bruk, enten endringen er en ny prompt, et nytt verktøy eller en ny modell, og resultatet er en poengsum for hver sak, tatt vare på over tid, slik at en rettelse for én sak ikke i det stille kan ødelegge en annen. Når en ny modell kommer, trengs det bare en endring på én linje i `routing.yaml` og en kjøring mot sakene for å prøve den ut, og beslutningen om å bytte hviler på poengsummene og ikke på kunngjøringen.

Slik beskriver dette nettstedet sine skreddersydde AI-agenter for din bransje. Valget av modell er konfigurasjon, ikke kode, med en liten og rask modell til sortering og uttrekk og en stor til vurderinger. Et oppdrag begynner med at tretti til femti reelle eksempler med riktige svar blir skrevet ned sammen med dem som gjør arbeidet, og hver endring, enten den gjelder en prompt, et verktøy eller en modell, kjøres mot alle sakene før den tas i bruk. Når en ny modell kommer, er det resultatet mot sakene og ikke kunngjøringen som avgjør om man bytter. Ifølge siden kan modellen byttes ut, og kunden eier koden, sakene og dataene, så agenten fortsatt er kundens når modellen under den byttes ut.

Hvor det går galt

Selvsikre svar som er feil. En modell som blir spurt om en kontraktsklausul den ikke har foran seg, kan finne på å beskrive en likevel, i den samme sikre tonen som ved et riktig svar, og kan vise til en kilde som ikke finnes. Gi modellen dokumentene den skal svare ut fra, be den vise til dem, og kontroller i kode at hver henvisning peker på et dokument og et avsnitt som finnes. Mål feilraten på nedskrevne saker, og la et menneske se gjennom svar som får konsekvenser, før noe handler på dem.

Instruksjoner gjemt i dataene. En applikasjon som lar en modell lese e-poster, nettsider eller opplastede filer, lar forfatterne av den teksten snakke til modellen. En artikkel fra 2023 viste dette mot ekte applikasjoner: instruksjoner plantet i innhold som modellen sannsynligvis ville lese, fikk den til å lekke data og endre hva applikasjonen gjorde, og NISTs profil nevner indirekte prompt injection blant sikkerhetsrisikoene ved generativ AI. Behandle alt modellen leser, som inndata du ikke kan stole på, håndhev tillatelser i verktøyene i stedet for i

prompten, gi verktøyene bare de rettighetene som personen modellen handler for, selv har, og krev at et menneske godkjenner enhver handling som ikke kan gjøres om.

En modell som endrer seg under deg. En ny versjon eller en annen leverandør kan svare annerledes på samme inndata, og et svar som besto forrige måned, feiler i dag, uten at din egen kode er endret. Angi en eksakt modellversjon der leverandøren tilbyr det, og kjør hele saksfilen ved hver endring, også et modellbytte, før den når brukerne.

Utdata som koden ikke kan lese. En modell som blir bedt om JSON, returnerer JSON fulgt av en vennlig setning, utelater et felt eller skriver et tall som en tekststreng. Beskriv den forventede formen i JSON Schema, kontroller hvert svar mot den, og behandle et svar som ikke består, som en feil som skal prøves på nytt eller sendes videre til et menneske, aldri som data.

Personopplysninger i prompter og logger. Alt som sendes til en modell hos en leverandør, forlater dine egne systemer, og en logg over prompter og svar tar vare på personopplysninger lenge etter at kallet er ferdig. NIST legger til at modeller kan gjenta sensitivt materiale fra treningsdataene sine, et problem som har blitt omtalt som memorering av data. Send bare feltene en oppgave trenger, hold promptloggene under de samme reglene for oppbevaring og tilgang som dataene i dem, og les leverandørens vilkår om oppbevaring og trening før personopplysninger når fram til den.

Les mer

- NIST AI 600-1: Generative Artificial Intelligence Profile — NISTs tillegg fra juli 2024 til sitt AI Risk Management Framework: tolv risikoer ved generativ AI, konfabulering blant dem, og foreslåtte tiltak for hver av dem.
- Attention Is All You Need — artikkelen fra 2017 som introduserte Transformer, som dagens språkmodeller bygger på.
- Language Models are Few-Shot Learners — artikkelen fra 2020 om en modell med 175 milliarder parametere som tar fatt på nye oppgaver ut fra eksempler i inndataene.
- Training language models to follow instructions with human feedback — finjustering på nedskrevne svar og rangerte utdata, steget fra tekstforutsier til assistent.
- Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection — instruksjoner plantet i data som en modell leser, demonstrert mot ekte applikasjoner.

Ordliste

Velg ut fra deres faktiske rammer

Beskriv arbeidsflyten, eksisterende grensesnitt og brukerne som trenger endringen. Vi kan definere en testbar leveranse og en overlevering som gjør teamet i stand til å vedlikeholde løsningen.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er en stor språkmodell?» 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/large-language-model/