Engineer

Stor sprogmodel

💬 et neuralt netværk, der forudsiger næste token i en tekst

Hvad det er

En stor sprogmodel er et neuralt netværk, der er trænet på en meget stor mængde tekst til at gøre én ting: ud fra teksten indtil nu at give hvert token, der kunne komme som det næste, en sandsynlighed. Et token er et kort stykke tekst, ofte et helt ord, nogle gange en del af et ord eller et enkelt tegn. At skrive et svar er det ene trin gentaget i en løkke. Programmet vælger et token ud fra sandsynlighederne, sætter det bag på teksten og spørger igen, indtil modellen giver et token, der betyder stop, eller en længdegrænse er nået. NIST's profil for generativ AI, udgivet i juli 2024, siger det på én linje: store sprogmodeller forudsiger det næste token eller ord i en sætning. Forkortelsen LLM, efter det engelske large language model, betyder det samme, og det er den form, profilen selv bruger på engelsk.

Arkitekturen bag de senere års modeller hedder Transformer og blev beskrevet i en artikel om maskinoversættelse fra 2017. Den droppede de rekurrente og konvolutionelle lag, som tidligere modeller var bygget af, og byggede udelukkende på attention, en mekanisme, der lader hver position i teksten se direkte på alle de andre. Det gjorde træningen langt lettere at fordele over mange processorer, og artiklens model blev trænet på væsentlig kortere tid end sine forgængere.

»Stor« henviser til antallet af parametre, de vægte, netværket lærer under træningen. En artikel fra 2020 beskrev en model med 175 milliarder af dem, ti gange flere end nogen tidligere ikke-sparse sprogmodel, og viste, at en model af den størrelse kunne tage fat på en ny opgave ud fra en håndfuld gennemarbejdede eksempler i sit input, helt uden yderligere træning.

En model, der kun er trænet til at fortsætte tekst, gør netop det: den fortsætter teksten, og det er ikke det samme som at gøre det, den blev bedt om. En artikel fra 2022 beskrev det trin, der gør sådan en model til en assistent. Mennesker skrev de svar, de ønskede, på et sæt anmodninger, og modellen blev finjusteret på dem. Derefter rangerede mennesker modellens egne svar, og den blev trænet videre på de rangeringer, en metode, der kaldes reinforcement learning from human feedback. I artiklens evalueringer foretrak folk svarene fra en model med 1,3 milliarder parametre, trænet på den måde, frem for svarene fra den oprindelige model med hundrede gange så mange parametre.

Brug begrebet i en beslutning

Brug »Hvad er en stor sprogmodel?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. En agent koblet til de systemer, I bruger, testet på sager, jeres egne eksperter skriver, under opsyn af jeres folk og aldrig låst til én model.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har bygget en kontrolmotor med 268 tjek — 268 commit-tjek og 146 tests af selve tjekkene

Har bygget en kontrolmotor med 268 tjek Kontakt os →

Hvorfor det betyder noget

En sprogmodel kan tage sig af arbejde, som er svært at dække med faste regler: at lægge en e-mail skrevet i fritekst i den rigtige mappe, at trække felterne ud af en scannet følgeseddel, at skrive et udkast til et svar på kundens eget sprog, at opsummere en lang tråd. Arbejde, der før krævede, at et menneske læste hver eneste sag, kan nu laves som udkast af software og tjekkes af et menneske.

Den egenskab, der gør den nyttig, er også det, der gør den svær at bygge på. Dens output bliver som regel trukket som en stikprøve, så det samme input kan give forskellige svar fra kørsel til kørsel, og den formulerer sig lige flydende, uanset om den har ret. NIST kalder selvsikkert fremført, forkert output for konfabulation, bedre kendt som hallucination, og beskriver det som en naturlig følge af, hvordan modellerne er bygget: de producerer tekst, der passer til statistikken i den tekst, de blev trænet på, og den er ofte rigtig og nogle gange ikke. NIST bemærker også, at en model kan konfabulere den begrundelse eller de kildehenvisninger, der ser ud til at underbygge et svar. Et forkert svar ligner et rigtigt til forveksling, så den eneste måde at vide, hvor ofte en model har ret i dit arbejde, er at måle den på dit arbejde.

Der er heller ikke én model at binde sig til. Nye modeller udkommer jævnligt, to af dem kan opføre sig ret forskelligt på det samme input, og det er udbyderen af den model, en applikation kalder, der bestemmer, hvornår den ændrer sig. Et team, der bygger på en sprogmodel, bygger på en del, det ikke selv styrer, og det arbejde, der holder, er alt det omkring modellen: de sager, der måler den, de værktøjer, den må kalde, og reglerne for, hvad den må gøre på egen hånd.

NIST's profil nævner tolv risikoområder for generativ AI, blandt dem konfabulation, databeskyttelse, informationssikkerhed og intellektuel ejendomsret. Den er skrevet til organisationer, der styrer risiko, snarere end til udviklere, men den fungerer fint som tjekliste for et team, der skal afgøre, hvad en funktion bygget på en model må røre ved.

Sådan ser det ud i praksis

En funktion bygget på en sprogmodel begynder med nedskrevne sager, ikke med en prompt. Hver sag er et rigtigt input og det svar, som en person, der kender arbejdet, ville godkende, gemt i en fil under versionsstyring ved siden af 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 model der tager hvilken slags opgave, er en indstilling, så en lille, hurtig model kan stå for udtrækket, mens en større skriver udkast til de svar, der kræver 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 ændring kører mod alle sagerne, før den går i drift, hvad enten ændringen er en ny prompt, et nyt værktøj eller en ny model, og resultatet er en score for hver sag, gemt over tid, så en rettelse for én sag ikke i det stille kan ødelægge en anden. Når en ny model udkommer, kræver det kun en ændring på én linje i `routing.yaml` og en kørsel mod sagerne at afprøve den, og beslutningen om at skifte hviler på scorerne og ikke på pressemeddelelsen.

Det er sådan, dette site beskriver sine skræddersyede AI-agenter. Valget af model er konfiguration, ikke kode, med en lille, hurtig model til sortering og udtræk og en stor til vurderinger. Et samarbejde begynder med, at tredive til halvtreds rigtige eksempler bliver skrevet ned med de korrekte svar, og hver ændring, hvad enten den gælder en prompt, et værktøj eller modellen, afprøves på dem alle, før den går i drift. Ifølge siden kan modellen skiftes ud, og kunden ejer koden, sagerne og dataene, så agenten fortsat er kundens, også når modellen bag den bliver udskiftet.

Hvor det går galt

Selvsikre svar, der er forkerte. En model, der bliver spurgt om en kontraktklausul, den ikke har foran sig, kan finde på at beskrive en alligevel, i samme sikre tone som ved et korrekt svar, og kan henvise til en kilde, der ikke findes. Giv modellen de dokumenter, den skal svare ud fra, bed den henvise til dem, og tjek i kode, at hver henvisning peger på et dokument og et afsnit, der findes. Mål fejlraten på nedskrevne sager, og lad et menneske se svar med konsekvenser, før noget handler på dem.

Instruktioner gemt i dataene. En applikation, der lader en model læse e-mails, websider eller uploadede filer, lader forfatterne af den tekst tale til modellen. En artikel fra 2023 viste det mod rigtige applikationer: instruktioner, der var plantet i indhold, modellen sandsynligvis ville læse, fik den til at lække data og ændre, hvad applikationen gjorde, og NIST's profil nævner indirekte prompt injection blandt sikkerhedsrisiciene ved generativ AI. Behandl alt, hvad modellen læser, som input, der ikke er til at stole på, håndhæv rettigheder i værktøjerne i stedet for i prompten, giv værktøjerne kun de rettigheder, som personen, modellen handler for, selv har, og kræv et menneskes godkendelse af enhver handling, der ikke kan gøres om.

En model, der ændrer sig under dig. En ny version eller en anden udbyder kan svare anderledes på det samme input, og et svar, der bestod i sidste måned, fejler i dag, uden at din egen kode er ændret. Angiv en præcis modelversion, hvor udbyderen tilbyder det, og kør hele sagsfilen ved hver ændring, et modelskift medregnet, før den når brugerne.

Output, som koden ikke kan læse. En model, der bliver bedt om JSON, returnerer JSON efterfulgt af en venlig sætning, udelader et felt eller skriver et tal som en tekststreng. Beskriv den forventede form i JSON Schema, tjek hvert svar mod den, og behandl et svar, der ikke består, som en fejl, der skal forsøges igen eller gives videre til et menneske, aldrig som data.

Persondata i prompter og logge. Alt, hvad der sendes til en model hos en udbyder, forlader dine egne systemer, og en log over prompter og svar gemmer persondata længe efter, at kaldet er afsluttet. NIST tilføjer, at modeller kan gentage følsomt materiale fra deres træningsdata, et problem, der er blevet kaldt memorering af data. Send kun de felter, en opgave har brug for, hold promptloggene under de samme regler for opbevaring og adgang som dataene i dem, og læs udbyderens vilkår om opbevaring og træning, før der når persondata frem til den.

Læs mere

- NIST AI 600-1: Generative Artificial Intelligence Profile — NIST's tillæg fra juli 2024 til sit AI Risk Management Framework: tolv risici ved generativ AI, konfabulation blandt dem, og forslag til handlinger for hver af dem.
- Attention Is All You Need — artiklen fra 2017, der introducerede Transformer, arkitekturen bag nutidens sprogmodeller.
- Language Models are Few-Shot Learners — artiklen fra 2020 om en model med 175 milliarder parametre, der tager fat på nye opgaver ud fra eksempler i sit input.
- Training language models to follow instructions with human feedback — finjustering på nedskrevne svar og rangerede output, skridtet fra tekstforudsiger til assistent.
- Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection — instruktioner plantet i data, som en model læser, demonstreret mod rigtige applikationer.

Ordliste

Vælg ud fra jeres faktiske vilkår

Beskriv arbejdsgangen, de eksisterende grænseflader og de brugere, der har brug for ændringen. Vi kan definere en testbar leverance og en overdragelse, der gør teamet i stand til at vedligeholde løsningen.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er en stor sprogmodel?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

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

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/large-language-model/