Engineer

Stor språkmodell

ett neuralt nätverk som förutsäger nästa token i en text

Vad det är

En stor språkmodell är ett neuralt nätverk som har tränats på en mycket stor mängd text för att göra en enda sak: att utifrån texten hittills ge varje token som kan komma härnäst en sannolikhet. En token är en kort textbit, ofta ett helt ord, ibland en del av ett ord eller ett enskilt tecken. Att skriva ett svar är samma steg upprepat i en slinga. Programmet väljer en token utifrån sannolikheterna, lägger den sist i texten och frågar igen, tills modellen ger en token som betyder stopp eller en längdgräns nås. NIST:s profil för generativ AI, utgiven i juli 2024, säger det på en rad: stora språkmodeller förutsäger nästa token eller ord i en mening. Förkortningen LLM, efter engelskans large language model, betyder samma sak, och det är den form som profilen själv använder på engelska.

Den uppbyggnad som de senaste årens modeller vilar på heter Transformer och beskrevs i en artikel om maskinöversättning från 2017. Den gjorde sig av med de rekurrenta och konvolutionella lager som tidigare modeller var byggda av och byggde enbart på attention, en mekanism som låter varje position i texten titta direkt på alla de andra. Det gjorde träningen mycket lättare att fördela över många processorer, och artikelns modell tränades på betydligt kortare tid än föregångarna.

”Stor” syftar på antalet parametrar, de vikter som nätverket lär sig under träningen. En artikel från 2020 beskrev en modell med 175 miljarder av dem, tio gånger fler än någon tidigare icke-gles språkmodell, och visade att en modell av den storleken kunde ta sig an en ny uppgift utifrån en handfull genomarbetade exempel i sina indata, helt utan vidare träning.

En modell som bara har tränats på att fortsätta text gör just det: den fortsätter texten, och det är inte samma sak som att göra det den blev ombedd att göra. En artikel från 2022 beskrev steget som gör en sådan modell till en assistent.

Människor skrev de svar de ville ha på en uppsättning förfrågningar, och modellen finjusterades på dem. Sedan rangordnade människor modellens egna svar, och den tränades vidare på rangordningarna, en metod som kallas reinforcement learning from human feedback. I artikelns utvärderingar föredrog människor svaren från en modell med 1,3 miljarder parametrar, tränad på det sättet, framför svaren från den ursprungliga modellen med hundra gånger så många parametrar.

Använd begreppet i ett beslut

Använd ”Vad är en stor språkmodell?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. En agent kopplad till de system du redan använder, testad på fall som dina egna experter skriver, övervakad av dina medarbetare och aldrig låst till en enda modell.

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. Byggde repots kontrollmotor med 268 kontroller — 268 commit-kontroller, och 146 tester av själva kontrollerna

Byggde repots kontrollmotor med 268 kontroller Hör av er →

Varför det spelar roll

En språkmodell kan ta sig an arbete som är svårt att täcka med fasta regler: att sortera ett mejl skrivet i fritext till rätt mapp, att hämta fälten ur en inskannad följesedel, att skriva ett utkast till svar på kundens eget språk, att sammanfatta en lång tråd. Arbete som tidigare krävde att en människa läste varje enskilt ärende kan nu förberedas som utkast av ett program och kontrolleras av en människa.

Samma egenskap som gör modellen användbar gör den också svår att bygga på. Utdata dras som regel slumpmässigt ur en sannolikhetsfördelning, så samma indata kan ge olika svar från körning till körning, och modellen formulerar sig lika flytande oavsett om den har rätt eller inte. NIST kallar säkert framförda, felaktiga utdata för konfabulering, mer känt som hallucination, och beskriver det som en naturlig följd av hur modellerna är byggda: de genererar text som passar statistiken i texten de tränades på, och den är ofta rätt och ibland inte. NIST påpekar också att en modell kan konfabulera den motivering eller de källhänvisningar som ser ut att stödja ett svar. Ett felaktigt svar ser förvillande likt ut som ett riktigt, så det enda sättet att veta hur ofta en modell har rätt i ditt arbete är att mäta den på ditt arbete.

Det finns inte heller en enda modell att binda sig till. Nya modeller kommer regelbundet, två av dem kan bete sig ganska olika på samma indata, och det är leverantören av den modell som ett program anropar som bestämmer när den ändras. Ett team som bygger på en språkmodell bygger på en del som det inte själv styr, och det arbete som består är allt runt modellen: fallen som mäter den, verktygen den får anropa och reglerna för vad den får göra på egen hand.

NIST:s profil nämner tolv riskområden för generativ AI, bland dem konfabulering, skydd av personuppgifter, informationssäkerhet och immateriella rättigheter. Den är skriven för organisationer som hanterar risker, mer än för utvecklare, men den fungerar väl som checklista för ett team som ska avgöra vad en funktion byggd på en modell får röra.

Så ser det ut i praktiken

En funktion byggd på en språkmodell börjar med nedskrivna fall, inte med en prompt. Varje fall består av riktiga indata och det svar som en person som kan arbetet skulle godkänna, och fallen ligger i en fil under versionshantering bredvid 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
```

Vilken modell som tar vilken typ av uppgift är en inställning, så en liten och snabb modell kan stå för extraktionen medan en större skriver utkast till de svar som kräver omdöme:

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

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

Varje ändring körs mot alla fall innan den släpps, oavsett om ändringen är en ny prompt, ett nytt verktyg eller en ny modell, och resultatet är en poäng per fall, sparad över tid, så att en rättelse för ett fall inte i det tysta kan förstöra ett annat. När en ny modell kommer behövs bara en ändring på en rad i `routing.yaml` och en körning mot fallen för att prova den, och beslutet att byta vilar på poängen och inte på lanseringen.

Så beskriver den här webbplatsen sina skräddarsydda AI-agenter för din bransch. Valet av modell är konfiguration, inte kod, med en liten, snabb modell för att sortera och extrahera och en stor för omdöme. Ett uppdrag börjar med trettio till femtio riktiga exempel med rätt svar, nedskrivna tillsammans med dem som gör arbetet, och varje ändring, vare sig det gäller en prompt, ett verktyg eller en modell, körs mot alla fallen innan den släpps. När en ny modell släpps är det resultatet mot fallen och inte lanseringen som avgör om man byter. Enligt sidan hålls modellen utbytbar, och kunden äger koden, fallen och datan, så att agenten förblir kundens när modellen under den byts ut.

Var det går fel

Självsäkra svar som är fel. En modell som får en fråga om en avtalsklausul den inte har framför sig kan hitta på en ändå, i samma säkra ton som vid ett riktigt svar, och kan hänvisa till en källa som inte finns. Ge modellen de dokument den ska svara utifrån, be den hänvisa till dem och kontrollera i kod att varje hänvisning pekar på ett dokument och ett stycke som finns. Mät felfrekvensen på nedskrivna fall, och låt en människa granska svar som får konsekvenser innan något agerar på dem.

Instruktioner gömda i data. Ett program som låter en modell läsa mejl, webbsidor eller uppladdade filer låter författarna till den texten tala till modellen. En artikel från 2023 visade detta mot verkliga applikationer: instruktioner som planterats i innehåll som modellen sannolikt skulle läsa fick den att läcka data och ändra vad applikationen gjorde, och NIST:s profil nämner indirekt prompt injection bland säkerhetsriskerna med generativ AI. Behandla allt modellen läser som indata du inte kan lita på, upprätthåll behörigheter i

verktygen i stället för i prompten, ge verktygen bara de behörigheter som personen modellen agerar för själv har, och kräv att en människa godkänner varje handling som inte kan göras ogjord.

En modell som ändras under dig. En ny version eller en annan leverantör kan svara annorlunda på samma indata, och ett svar som klarade sig förra månaden misslyckas i dag, utan att din egen kod har ändrats. Ange en exakt modellversion där leverantören erbjuder det, och kör hela fallfilen vid varje ändring, även ett modellbyte, innan den når användarna.

Utdata som koden inte kan läsa. En modell som ombeds svara med JSON returnerar JSON följt av en vänlig mening, utelämnar ett fält eller skriver ett tal som en textsträng. Beskriv den förväntade formen i JSON Schema, kontrollera varje svar mot den och behandla ett svar som inte klarar kontrollen som ett fel som ska försökas igen eller skickas vidare till en människa, aldrig som data.

Personuppgifter i prompter och loggar. Allt som skickas till en modell hos en leverantör lämnar dina egna system, och en logg över prompter och svar bevarar personuppgifter långt efter att anropet är klart. NIST tillägger att modeller kan återge känsligt material ur sina träningsdata, ett problem som har beskrivits som memorering av data. Skicka bara de fält som en uppgift behöver, håll promptloggarna under samma regler för bevarande och åtkomst som uppgifterna i dem, och läs leverantörens villkor om bevarande och träning innan personuppgifter når den.

Läs mer

- NIST AI 600-1: Generative Artificial Intelligence Profile — NIST:s tillägg från juli 2024 till sitt AI Risk Management Framework: tolv risker med generativ AI, konfabulering bland dem, och föreslagna åtgärder för var och en.
- Attention Is All You Need — artikeln från 2017 som introducerade Transformer, som dagens språkmodeller bygger på.
- Language Models are Few-Shot Learners — artikeln från 2020 om en modell med 175 miljarder parametrar som tar sig an nya uppgifter utifrån exempel i indata.
- Training language models to follow instructions with human feedback — finjustering på nedskrivna svar och rangordnade utdata, steget från textförutsägare till assistent.
- Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection — instruktioner planterade i data som en modell läser, demonstrerade mot verkliga applikationer.

Ordlista

Välj utifrån era faktiska förutsättningar

Beskriv arbetsflödet, befintliga gränssnitt och användarna som behöver förändringen. Vi kan definiera en testbar leverans och en överlämning som gör att teamet kan underhålla lösningen.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är en stor språkmodell?” 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/large-language-model/