Förslag Till Riktlinjer För Användandet Av Stora Språkmodeller Inom Sjukvården

GPT-Internship 2023, AI Sweden

Pauli, Oskar
oskar.pauli@affiliate.ai.se

Johansson, Henrik
henrik.johansson@affiliate.ai.se

Nilsson, Felix
felix.nilsson@affiliate.ai.se

August 9, 2023

Contents

1 Introduktion

Detta dokument är ämnat som ett förslag på riktlinjer för implementationen av stora språkmodeller inom sjukvården. Riktlinjerna är baserade på författarnas erfarenheter under sommaren 2023, där de utvecklade en demoapplikation som använde stora språkmodeller inom sjukvården.

1.1 Författarna

Dessa riktlinjer är skrivna av Henrik Johansson, Felix Nilsson och Oskar Pauli under deras internship på AI-Sweden sommaren 2023. Alla tre är masterstudenter på Chalmers Tekniska Högskola med respektive huvudområde inom mjukvaruutveckling, datavetenskap samt matematik. Det kan därför vara värt att notera att de är skrivna från ett tekniskt perspektiv i första hand, snarare än ett medicinskt.

1.2 Bakgrund

Stora språkmodeller, (Engelska: *Large Language Models*, LLMs) är ett samlingsnamn för olika typer av program som framförallt använder sig av djup maskininlärning för att utveckla en förståelse (t.ex syntax eller andra samband) för språk. Detta görs genom träning på ett eller flera *korpusar*, större samlingar av text. För bästa prestanda används vanligen flera olika korpusar för att täcka ett brett spann av ämnen och språkanvändning.

LLMs har sina rötter i statistisk maskinöversättning vilket tog sin början runt 1900-talets mitt, och blev ett tidigt användningsområde för djupa maskininlärningstekniker i form av neurala nätverk under 2010-talet. Utvecklingen av LLMs genomgick dramatisk förbättring 2017 då den inflytelserika artikeln "Attention is all you need" av Vaswani et al. [11] introducerade transformer-arkitekturen, vilket lättare kunde parallelliseras jämfört med samtida arkitekturer och därmed innebar att LLMs kunde tränas mer effektivt. Sedan dess utgör transformen grunden för merparten av moderna LLMs.

Hur omfattande en LLM är mäts oftast i hur många *parametrar* den använder, där fler (oftast [6]) innebär att modellen bättre kan fånga de många nyanserna inom språkbruk. Fler parametrar introducerar dock en större börda under träningsfasen, vilket har resulterat i utveckling av de allra största modellerna är reserverade för ett fåtal företag med stora resurser. I praktiken är en parameter ett flyttal som förvaras t.ex i en matris, vilken uppdateras under träningsfasen. Forskning har nyligen visat att dessa tal kan skalas ner från t.ex 16-bitars precision till så låg som 3-bitars precision och fortfarande bibehålla goda resultat, vilket kan visas viktigt för att köra modellerna på sämre hårdvara [5].

1.3 Modeller Idag

Moderna LLMs har idag nått en oerhörd storlek i antal parametrar, vilket vida överstiger vad till exempel en persondator kan erbjuda för att köras lokalt. Från ett blogginlägg på huggingface [7] rekommenderas att t.ex Llama 2-70b (vilket beskrivs i detalj längre ner) bör köras på åtminstone två Nvidia A100 GPU:er, vilka kostar kring åtminstone 200 000 kr styck. Detta kan kringås genom vissa molntjänster som t.ex Google Colab som erbjuder användning av Nvidia A100 mot en avgift. Det bör nämnas att vissa företag även erbjuder mindre varianter av sina modeller, som kan köras på sämre hårdvara som t.ex mobiltelefoner.

De företag som utvecklar LLMs erbjuder därför oftast tillgång via ett API vilket i kontexten av sjukvård kan vara problematiskt då potentiellt känslig data kommer behandlas av utomstående företag. Det innebär även att modellens anpassningsbarhet är begränsad till som mest *few-shot learning* men inte fulländad *fine-tuning*.

Den kanske mest kända modellen är GPT-3 som driver ChatGPT, och består av 175 miljarder parametrar [4]. Uppföljaren GPT-4 har inga officiella siffror i skrivande stund, men har påståtts ha ca 1000 miljarder parametrar [1] vilket skulle göra den till den överlägset största modellen. OpenAIs modeller är öppna för användning via deras API, men inte lokalt.

Ett annat företag som spenderar en del resurser på området är Google. Deras senaste modell PalM-2 utgör basen för deras konkurrent till ChatGPT, Bard, men har även visats prestera väl för mer specifika uppgifter. Relevant till sjukvården är att en variant av PalM-2 utgör den just nu bäst presterande medicinska frågeställningsmodellen [9], vilket presterar i nivå med läkare på olika medicinska quizer. PalM-2 använder 10 miljarder parametrar [2] och är varken öppen via ett API eller lokalt i skrivande stund.

Till sist bör även företaget Meta nämnas, vars senaste modell, Llama-2, nyligen avtäcktes [10]. Vad som särskiljer den från andra tidigare modeller är att den är fullständigt *open-source* och den fullständiga 70 miljarder parametrar stora modellen finns tillgänglig för nedladdning och kommersiellt bruk. Medan GPT-4 fortfarande konkurerar ut den som en generell chatbot, så anser vi att Llama-2 har en hel del potential som den enda stora språkmodellen som kan köras fullständigt lokalt och fine-tuneas till ett användningsområde som medicin.

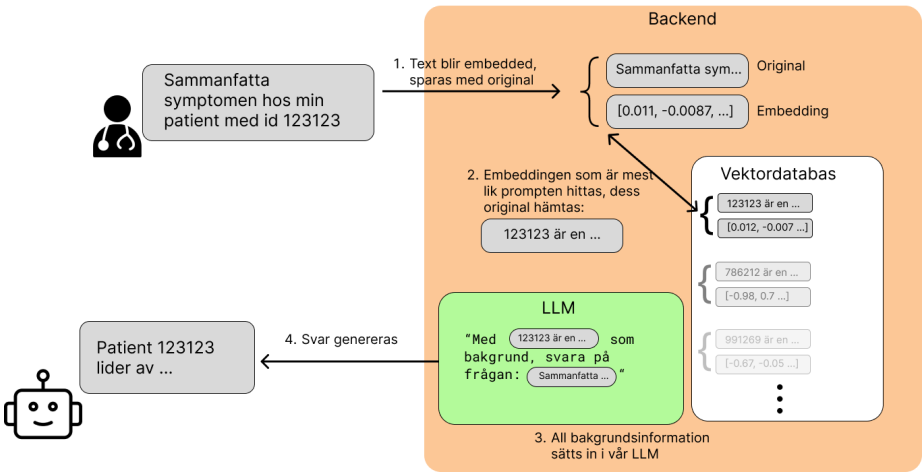
2 Riktlinjer

2.1 Korrekthet

Som nämnt ovan är LLMs statistiska objekt, och det är lämpligt att alltid agera utifrån att de kan göra fel som att påstå något som inte stämmer, eller ge ett partiskt svar. Dessa typer av felaktiga svar kallas för att modellen *hallucinerar* och härrör från den partiskhet som existerar inom de korpusar som modellen tränas på. Det är inte nödvändigtvis ett resultat av dåliga val av träningsdata, då forskning har visat att även väldigt noga kurerade dataset kan leda till partiska modeller [3]. [1]

I kontexten av sjukvård föreslår vi att användning av LLMs sker tillsammans med källor, och ger svar med dessa som bakgrund. Källor kan t.ex vara filer i format som `PDF`, `JSON` eller `ICS` (de bör vara i ett s.k. *human-readable* (HR) format, eller läsas in som HR innan de används) och kan effektivt sättas in som kontext till en modell. Detta kan förslagsvis ske genom så kallad *similarity search*, där en fråga eller *prompt* jämförs med sparade dokument och det mest liknande dokumentet blir utvalt som bakgrundsinformation till vår LLM. Ett exempel på hur en sådan chatbot kan struktureras kan ses härunder, där en doktor kan ställa en LLM frågor om sina patienter:

Översikt över ett exempel på chatbotsarkitektur inom sjukvård



[1] Ifall du som läsare bättre vill förstå teorin bakom similarity search (bl.a koncept som embeddings och vektoravstånd), så kan du läsa följande artikel: What is similarity search

Det bör finnas begränsningar på sökområdet som t.ex att om användaren är en patient, bör inte hen kunna ställa frågor på dokument tillhörande en annan patient. Det är även värt att notera att sådan begränsning kan vara väldigt hjälpsam, då det försäkrar att bakgrundsdokumenten håller en hög relevans till kontexten.

Utöver den väldigt förenklade bilden finns det en hel del strategier som i praktiken kan ge fördelar inom både prestanda och kvalitet. Nedan följer några exempel:

Chunking

Chunking innebär att dokument bryts ner och sparas i mindre delar vilket är mer effektivt, då en LLMs kontextfönster generellt inte kan använda (eller i många fall använda på ett effektivt sätt) större än några sidor av kontinuerlig text. Då kan man t.ex använda flera chunks över olika dokument istället för ett monolitiskt dokument, vilket kan ge ett bättre svar. Detta kan visa sig svårt i verkligheten, då vissa format som `JSON` kan vara lätta att dela upp i oberoende chunks, medan format som text kan refera fram och tillbaka till andra delar av text och chunksen kan lätt förlora sin mening. Detta problem kallas för "content aware chunking" [8].

Vektordatabaser

En vektordatabas är en databas som snabbt och effektivt kan spara innehållet hos dokument tillsammans med dess motsvarande embedding. Detta mer eller mindre eliminerar behovet att "för hand" skapa och uppdatera embeddings och är ett smidigt verktyg om man vill implementera similarity search. Det finns många alternativ, men två vanliga databaser är Pinecone och Chroma.

2.2 Ton & Språkval

En LLM-baserad applikation kommer främst att interagera med användaren genom text, t.ex i form av en chat. Därför är det viktigt att specificera i systemmedelandet hur informationen ska presenteras. Ifall det t.ex inte specificeras vilket språk som bör användas kommer svaret antagligen komma på samma språk som frågan från användaren. Det kan å ena sidan vara en fördel för att möjliggöra applikationen för fler än bara svensktalande, men kan å andra sidan försämra t.ex similarity searchen (det kan innebära att frågan behöver preprocessas och översättas till språket som används i vektordatabasen).

Det är även viktigt att specificera vilken språknivå som modellen bör använda. För interaktioner med en läkare kan det vara passande med en teknisk ton, medan för en generell vårdsökande kan det vara lämpligt att presentera informationen på ett lättillgängligt sätt.

Vidare kan det vara värdefullt att specificera om modellen får spekulera om saker som behandling, eller ifall sådana rekommendationer bör lämnas till en t.ex en läkare

2.3 Patientdata

Det största hindret som vi ser just nu är hur man ska respektera patientens data som är sekretessbelagd när man använder LLMs. Det finns just nu ingen möjlighet att köra de bästa modellerna lokalt i varje region, vilket innebär att data kommer skickas någon annanstans, något som inte är tillåtet.

Vad som också är viktigt att tänka på ur ett patientsäkerhetsperspektiv är att vara nogrann med att rätt personer har tillgång till rätt dokument. Om språkmodellen får tillgång till någon annan patients journaler bryter vi dels mot patientdatalagen men svaret som ges kan också vara felaktigt.

2.4 Testning

Testning av applikationer med LLMs kan vara en utmaning, då till och med små ändringar till systemmeddelandet kan ha oförutsedda konsekvenser. Som nämnt kan det inträffa att modellen uppvisar vad som verkar vara försämrad kvalitet i sina svar, men i själva verket var ett ytterligghetsfall. Trots dessa bekymmer så finns det en del resurser att tillgå för att systematiskt testa en LLM-applikation.

Vi valde en typ av testning som går ut på att man skapar en fråga med ett eller flera kända "rätta" svar. Vi ber därefter modellen att svara på frågan och jämför sedan hur likt kandidatsvaret var med referenssvaren. Den likheten kan bland annat mätas på samma sätt som nämndes i avsnittet om korrekthet, dvs med embeddings. Ett alternativ som vi såg presterade väl var att initiera en ny instans av en LLM och be den bedömma ifall svaren var lika, varpå den svarar med ett ja eller nej. Dessa resultat kan sedan samlas ihop och summeras som t.ex ett stapeldiagram.

Vi ser testning av LLM-applikationer som lika viktigt som testning av traditionell mjukvara. Testning borde ske kontinuerligt under utvecklingen och testfall borde täcka ett brett spann av interaktioner för att försäkra att modellen inte tappat i kvalitet. Med detta i åtanke så kan det vara värt att överväga något av de existerande ramverk för testning, och där fann vi att promptfoo var ett relativt välfungerande verktyg. Det är dock viktigt att tänka på att landskapet för LLM-utveckling förändras snabbt, och det kan finnas andra bra alternativ.

2.5 Avgränsningar

Språkmodeller är väldigt breda och kan väldigt mycket, därför är det extra viktigt att avgränsa sin applikation och ha ett tydligt syfte med vad den ska göra. Exempelvis kan man inom sjukvården vilja ha en chattbot som kan söka upp och citera information, men om den också börjar ge medicinska råd i sitt svar blir det problematiskt i och med att detta skulle kunna influera läkarens bedömning. Detta hade i sin tur inneburit att chattboten hade klassats som en medicinteknisk produkt och med det tillkommer högre krav och säkerhet, vilket

hindrar personal från att använda applikationen för det som den är tänkt och godkänd för.

Vi har hittat följande rimliga användningsområden för språkmodeller inom sjukvården:

- Patientinteraktion - Skapa en chattbot som kan prata med patienter och svara på frågor som patienten har angående sin journal och sjukhusvistelse.
- Läkarassistent - Skapa en chattbot som kan hjälpa läkare med att leta upp information om patienter snabbt.
- Transkribering och generering - Skapa ett program som kan sammanfatta transkriberade samtal och dikteringar.
- Administrativa uppgifter - Detta kan innebära att återigen skapa en chattbot där personal kan på ett enklare sätt hitta information i intranätet.
- Beslutsstöd - Använda GPT-modeller till att analysera journaler och samtal mellan patient och läkare för att exempelvis kunna ge rekommendationer på möjliga tester som kan göras. Detta är dock ett användningsområde som liger lite längre bort i tiden än de tidigare punkterna då denna applikation kan påverka en klinikers tillvägagångsätt, vilket klassar produkten som en medicinteknisk produkt (MTP). Detta höjer säkerhetskraven på produkten vilket måste tillfredsställas innan den kan användas.

2.6 Användaren

När det kommer till gränssnittet mellan användare och språkmodell bör man ha i åtanke vem användaren är. Om man, som i vårt fall, gör en chattbot som patienter kan skriva med måste man tänka på att alla typer av människor ska kunna använda den. Detta kan innebära att gränssnittet ska vara simpelt utan många inställningar, val m.m. Gränssnittet för en läkare kan å andra sidan vara mer avancerad där inställningar för olika typer av struktur och längd på svaren kan ge läkaren bättre hjälp.

3 Sammanfattning

Sammanfattningsvis föreslår dessa riktlinjer vad man bör tänka på vid användning av språkmodeller (LLMs) inom sjukvården för en ansvarsfull och effektiv implementering.

Användningsområdena för språkmodeller inom sjukvården är många och sträcker sig från administrativa uppgifter som att hitta och sammanfatta information, till att agera som en läkarassistent som kan användas som beslutsstöd och mycket annat.

För att effektivisera dokumenthantering kan "chunking" användas, vilket delar upp ett dokument i mindre delar så att språkmodellen bara tar in den relevanta

informationen för att svara på den ställda frågan. Vidare fann vi även att ett effektivt sätt att hitta rätt "chunk" är genom att göra en så kallad "similarity search".

Den mest centrala aspekten som lyfts är hanteringen av patientdata. Datan är sekretessbelagd och med det kommer problem som diskuteras i texten. Vidare nämns det även att väldefinierade användningsområden och ett användarvänligt gränssnitt är två andra aspekter som är viktiga för en lyckad produkt.

References

[1] Reed Albergotti. *The secret history of elon musk, Sam Altman, and openai*. Mar. 2023. URL: https://www.semafor.com/article/03/24/2023/the-secret-history-of-elon-musk-sam-altman-and-openai.

[2] Rohan Anil et al. *PaLM 2 Technical Report*. 2023. arXiv: 2305.10403 [cs.CL].

[3] Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Kalai. "Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings". In: *CoRR* abs/1607.06520 (2016). arXiv: 1607.06520. URL: http://arxiv.org/abs/1607.06520.

[4] Tom B. Brown et al. "Language Models are Few-Shot Learners". In: *CoRR* abs/2005.14165 (2020). arXiv: 2005.14165. URL: https://arxiv.org/abs/2005.14165.

[5] Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. *QuIP: 2-Bit Quantization of Large Language Models With Guarantees*. 2023. arXiv: 2307.13304 [cs.LG].

[6] Ian R. McKenzie et al. *Inverse Scaling: When Bigger Isn't Better*. 2023. arXiv: 2306.09479 [cs.CL].

[7] Philip Schmid, Omar Sanseviero, Pedro Cuenca, and Lewis Tunstall. *Llama 2 is here - get it on hugging face*. URL: https://huggingface.co/blog/llama2#using-text-generation-inference-and-inference-endpoints.

[8] Roie Schwaber-Cohen. *Chunking Strategies for LLM Applications*. https://www.pinecone.io/learn/chunking-strategies/. [Accessed 01-08-2023].

[9] Karan Singhal et al. *Towards Expert-Level Medical Question Answering with Large Language Models*. 2023. arXiv: 2305.09617 [cs.CL].

[10] Hugo Touvron et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: 2307.09288 [cs.CL].

[11] Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.