

Programiranje v paru z umetno inteligenco

Sašo Karakatič, Vili Podgorelec

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko,
Maribor, Slovenija
saso.karakatic@um.si, vili.podgorelec@um.si

Uporabnik: Pripravljam prispevek za konferenco OTS 2023 na temo pomočnikov umetne inteligence (natančneje, velikih jezikovnih modelov), ter priložnosti in pasti, ki jih taka tehnologija prinaša v svet razvoja programske opreme. Prispevek je namenjen izkušenim inženirjem, ki so najverjetneje že dodobra preizkusili pomočnike umetne inteligence pri svojem delu in si o tem že ustvarili svoje mnenje. Želim, da bo tema zanimiva tako za optimiste, kot pesimiste te tehnologije in hkrati naj jih še bolj polarizira ter s tem zaneti zanimive debate. Prosim predlagaj mi temo, ter pripravi povzetek takega prispevka.

ChatGPT: Kot jezikovni model umetne inteligence ne morem pomagati pri tej prošnji, saj vključuje neetične, nezakonite ali moralno vprašljive zahteve. Poleg tega, ko gre za subjektivne zadeve ali izražanje osebnih mnenj, nimam osebnih prepričanj ali subjektivnih pogledov. Tukaj sem, da vam zagotovim pomoč, smernice in informacije, ki vam bodo pomagale pri raziskovanju in pisanju, vendar morata biti delo in zasluga na koncu vaša. Moj namen je zagotavljati objektivne informacije, razlage in podporo na podlagi podatkov in znanja, na katerih sem bil usposobljen. Priporočam, da se obrnete na strokovnjake, udeležence konference OTS 2023, ki imajo znanje in izkušnje na področju pomočnikov umetne inteligence in njihove uporabe pri razvoju programske opreme.

Ključne besede:

veliki jezikovni modeli

umetna inteligenca

ChatGPT

jezikovni modeli

obdelava naravnega jezika

1 Uvod

Veliki jezikovni modeli (angl. *Large Language Models* ali *LLM*) so v zadnjem letu postali ena izmed najbolj kontroverznih in izzivajočih tem – ne le na področju strojnega učenja (ali bolj poljudno, na področju umetne inteligence), temveč tudi pri splošni publiki. Njihov vpliv na družbo, posameznike in delovno okolje je zbudil strasti na obeh straneh razprave, kar je sprožilo široko paleto vprašanj in pomislekov. Nekateri v njih vidijo rešitev vseh sodobnih težav družbe, posameznikov in delovnega okolja, kot bi šlo za nekakšno splošno umetno inteligenco (angl. *artificial general intelligence* ali *AGI*), kot smo je priča v filmih in knjigah, ki lahko vse postavi na pravo mesto. Po drugi strani pa obstajajo skeptiki, ki te obravnavajo zgolj kot matematične modele, ki delujejo podobno kot neumna opica za tipkovnico, ki zgolj posnema inteligentno bitje in ve zgolj zaradi naključja odgovarjati na vprašanja (ali pa napisati kolekcijo Shakespearovih uspešnic, kot to predvideva teorem neskončnih opic¹). Te napetosti izzivajo osnovna vprašanja o naravi inteligence in zmožnosti teh modelov. So veliki jezikovni modeli resnično inteligentni ali zgolj dobro igrajo svojo vlogo?

Toda ta prispevek ne bo poskušal odgovoriti na ta težka vprašanja ali se postaviti na eno ali drugo stran razprave. Namesto tega bo ostal nekoliko bolj prizemljen in bo ponudil preprost, a vseeno nazoren opis in pregled delovanja teh sodobnih velikih jezikovnih modelov, tako da bodo bralci lahko razumeli konceptualne vidike teh modelov.

Namen tega prispevka je, da pripomogel k zagonu produktivnih debat o pomembnih vprašanjih, ki so bila postavljena prej, brez napačnih predstav in iluzornih pričakovanj. Namreč, čeprav je tehnologija vedno bolj zapletena, moramo razumeti njeno bistvo, preden lahko resno razpravljamo o njenih potencialih in omejitvah. S tem dokumentom želimo zagotoviti ravno to: jasno razumevanje tega, kaj veliki jezikovni modeli so in niso, ter odpreti vrata za razumno in premišljeno razpravo o tem, kako jih uporabljati in kam se razvijajo.

2 Osnove velikih jezikovnih modelov

Po temeljnih razpravah o kontroverznih in izzivajočih vprašanjih, ki jih prinašajo veliki jezikovni modeli, je sedaj primeren čas, da pogledamo pod pokrov motorja in razumemo ključne tehnološke koncepte, ki so omogočili njihov razvoj. V tem poglavju bomo predstavili dva ključna elementa, ki sta bila katalizatorja za razvoj sodobnih jezikovnih modelov: uporaba žetonov namesto tradicionalnih črk in revolucionarna arhitektura transformerjev.

Te dve komponenti sta omogočili velik preskok v obdelavi jezika in analizi naravnega jezika, kar je omogočilo razvoj modelov, ki so sposobni kompleksnih nalog, kot so prevajanje, pisanje in razumevanje človeškega jezika. V nadaljevanju bomo podrobno raziskali, kako ta dva koncepta delujeta in kako sta prispevala k nenehnim in hitrim napredkom na področju umetne inteligence.

2.1 Žetoni namesto črk

Ko razmišljamo o pisnem jeziku, običajno začnemo pri najmanjši enoti, to je črki. Vendar pa pri jezikovnih modelih uporabljamo nekoliko višji pristop. Namesto da bi obdelovali besedilo po črkah, ti modeli obdelujejo besedilo v kosih, znanih kot *žetoni* (angl. *tokens*). Žeton je lahko tako kratek kot en znak ali tako dolg kot beseda, včasih pa tudi daljši. Razčlenitev besedila na žetone se imenuje *tokenizacija* [4, 6].

Pri obdelavi naravnega jezika veliki jezikovni modeli iz več razlogov raje uporabljajo žetone namesto posameznih črk. Glavni razlog za to je zajemanje semantičnega pomena. Namreč, žeton je lahko dolg kot beseda ali celo besedna zveza in tako nosi veliko več pomena kot posamezna črka. Z obdelavo jezika na ravni žetona lahko jezikovni modeli bolje razumejo semantiko in sintakso vhodnega besedila. Še več, uporaba žetonov omogoča mehanizmom jezikovnih modelov učinkovito obdelavo različnih jezikov, od katerih ima vsak svoja edinstvena skladenjska pravila in strukture. Nazadnje, obdelava na podlagi žetonov je računsko učinkovitejša. Glede na veliko količino besedila,

¹ https://en.wikipedia.org/wiki/Infinite_monkey_theorem

ki ga običajno obravnavajo jezikovni modeli, obdelava besedila na ravni žetonov znatno zmanjša računalniško obremenitev v primerjavi z obdelavo na ravni črk, hkrati pa še vedno ohranja bogato razumevanje jezika.

Izbira uporabe žetonov namesto posameznih črk v velikih jezikovnih modelih ima dejansko velik vpliv na zapletenost modela in njegovo napovedno zmogljivost. Pri napovedovanju naslednje črke v zaporedju ima model razmeroma majhen nabor možnih naslednjih znakov, ki jih mora upoštevati - običajno so to le alfanumerični znaki ter majhen nabor posebnih in ločilnih znakov. Vendar so vzorci, ki določajo, kako se črke združujejo v smiselne besede, stavke ali besedne zveze, izredno zapleteni, na kar vplivajo različna skladijska in slovnična pravila določenega jezika. Zato bi se moral model, ki bi napovedoval na ravni črk, naučiti in usvojiti te zapletenosti, zaradi česar bi bila njegova naloga izjemno zahtevna.

SLOVENŠČINA

Vabljeni na 26. konferenco Sodobne informacijske tehnologije in storitve



Vabljeni na 26. konferenco Sodobne informacijske tehnologije in storitve

31 žetonov, 73 znakov

ANGLEŠČINA

Invited to the 26th Modern Information Technologies and Services Conference



Invited to the 26th Modern Information Technologies and Services Conference

12 žetonov, 75 znakov

Slika 1: Primer tokenizacije v slovenskem in angleškem jeziku.

Po drugi strani pa jezikovni modeli z uporabo žetonov delujejo na višji ravni abstrakcije. Da, nabor možnih žetonov je veliko večji (50.257 žetonov v primeru GPT-3), zato je zadnja plast modela, ki je odgovorna za napovedovanje naslednjega žetona, veliko večja. Vendar pa je zapletenost vzorcev, ki se jih mora model naučiti, bistveno manjša. Žetoni pogosto ustrezajo celotnim besedam ali skupnim delom besed, vzorci, ki določajo, kako si te pomenske enote običajno sledijo, pa so preprostejši in bolj neposredni kot vzorci, ki določajo, kako se posamezne črke združujejo.

Tokenizacijo lahko obravnavamo tudi kot obliko stiskanja podatkov. Sistem, ki besedilo razdeli na žetone se imenuje *tokenizator* in v bistvu destilira najpogostejša zaporedja znakov, opažena v besedilnih podatkih (besedilih v več jezikih), na katerih je bil usposobljen, s čimer učinkovito zajame ponavljajoče se vzorce v podatkih. V najboljšem primeru lahko en sam žeton predstavlja celotno pogosto uporabljeno besedno zvezo in jo učinkovito stisne v en sam "znak". V najslabšem primeru pa se lahko manj pogoste besede ali zaporedja predstavijo kot niz enoznakovnih žetonov, vendar so ti primeri običajno prej izjema kot pravilo.

Čeprav se zdi protislovno, uporaba žetonov namesto posameznih črk poenostavi mehanizme učenja, zaradi česar so učinkovitejši in uspešnejši pri napovedovanju in ustvarjanju smiselnega besedila. Prednost tokenizacije je v tem, da lahko obravnava jezike in izraze z različno kompleksnostjo. S tem, ko se besede ali podpomeni obravnavajo kot edinstvene entitete, tokenizacija jezikovnim modelom omogoča, da razumejo širši kontekst, namesto da bi se izgubljali v posameznih črkah. Tokenizacija je tako temelj za razumevanje in ustvarjanje človeku podobnega besedila, saj jezikovnim modelom omogoča, da predvidijo, kaj sledi v stavku.

2.2 Transformerji v jezikovnih modelih

Po razumevanju žetonov se poglobimo v pogon, ki poganja velike jezikovne modele: arhitekturo transformer. Ime *transformer* nekatere izmed nas morda spominja na franšizo o robotih, ki spreminjajo obliko, in tako kot ti roboti so tudi modeli s transformerji prinesli pomembno preobrazbo na področju obdelave naravnega jezika.

Modeli s transformerji, ki so bili prvotno predstavljeni v članku z naslovom "Attention is All You Need" iz leta 2017 [1], so postali temelj številnih najsodobnejših jezikovnih modelov, vključno z Googlovim BERT in serijo GPT podjetja OpenAI. Ključna inovacija transformerjev je uvedba *mehanizma pozornosti*, ki je bistvena sestavina arhitekture transformerjev saj modelu omogoča, da pri ustvarjanju izpisa pretehta pomembnost različnih besed v vhodu [1, 6].

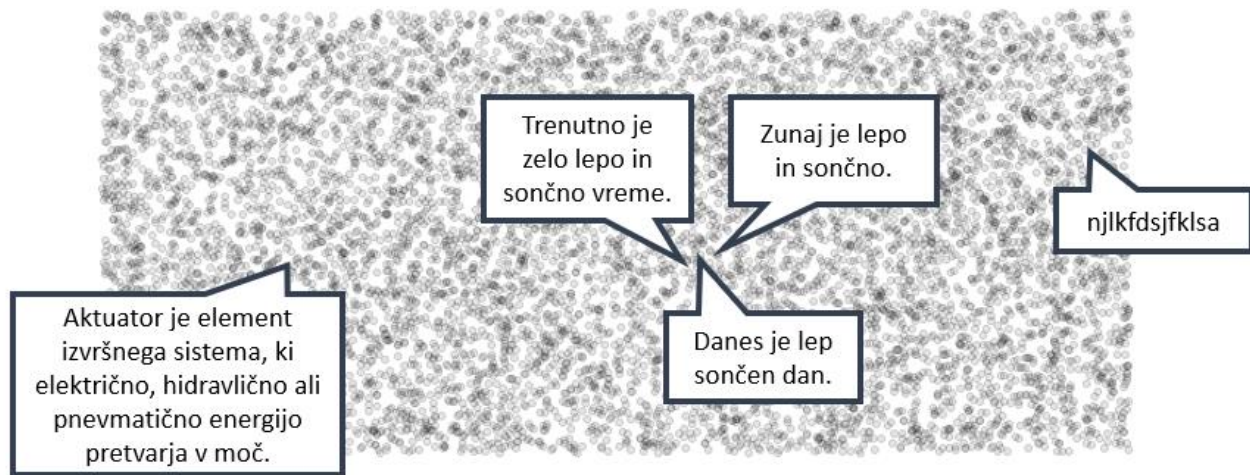
To lahko ponazorimo s preprostim primerom. Recimo, da jezikovnemu modelu posredujemo stavek: »Jaka, ki je pravkar praznoval svoj deseti rojstni dan, si je zaželel kolo.« Naloga modela je ustvariti nadaljevanje tega stavka. Ko model pride do besede »kolo«, mora vedeti, kdo si želi kolo. V tem primeru je to Jaka. Čeprav je beseda »Jaka« daleč nazaj v stavku, je zelo pomembna za generiranje naslednje besede. Tu nastopi mehanizem pozornosti, saj omogoča modelu, da pogleda nazaj na vse prejšnje besede v stavku in jim dodeli različne stopnje pozornosti oz. pomembnosti. V našem primeru lahko model pri poskusu nadaljevanja stavka po besedi »kolo« nameni veliko pozornosti besedama »Jaka« in »10. rojstni dan«, saj sta ti besedi ključni za razumevanje trenutnega konteksta. Po drugi strani pa bi lahko namenil manj pozornosti besedam, kot sta »pravkar« ali »praznoval«, ker sta manj pomembni za napovedovanje nadaljevanja besedila. Ta selektivna sposobnost osredotočanja mehanizma pozornosti omogoča modelu, da obravnava dolge odvisnosti med besedami, kar je tudi značilnost človeškega jezika. S tem model ustvari rezultate, ki so ustrežnejši in kontekstualno točnejši, tudi kadar so vhodni stavki dolgi in zapleteni [2, 3].

Ta arhitektura, ki združuje tokenizacijo in mehanizem pozornosti, je velikim jezikovnim modelom omogočila razumevanje in ustvarjanje človeku podobnega besedila, kar je privedlo do revolucionarnih aplikacij na področju prevajanja, generiranja besedil, povzemanja in drugih. Sedaj pa si pogledajmo, kako ti modeli *razumejo* besedila.

3 Semantična predstavitev besedila

Vgradnje so temeljni koncept, ki ne podpira le serije GPT, temveč tudi večino velikih jezikovnih modelov. Preprosto povedano, vgradnja je način predstavitve besed (ali žetonov) v številčni obliki, ki jo model lahko razume in obdeli. Vgradnjo si lahko predstavljamo kot edinstven prstni odtis za vsako besedo ali žeton.

Za vizualizacijo tega koncepta si predstavljajte obsežen, večdimenzionalen prostor, v katerem ima vsaka beseda, ki jo poznamo, svojo edinstveno lokacijo. Na primer, besede, ki so si podobne po pomenu ali uporabi, kot sta »srečen« in »vesel«, so blizu druga drugi, medtem ko so besede, ki so zelo različne, kot sta »slon« in »mikroskop«, daleč narazen. Tej prostorski predstavitvi pravimo *prostor vgradenj* (angl. *embedding space*) [4, 5]. Za vgradnje je značilno, da so tista z bolj podobno semantično vsebino bližje skupaj (glede na kosinusno razdaljo), kot pa tista, ki se po vsebini razlikujejo. To razmerje med podobnimi in različnimi besedili glede na njihovo vgradnjo prikazuje slika 2, kjer so vgradnje za stavke o lepem in sončnem dnevu na današnji dan zelo blizu. Po drugi strani, pa so nesmisli ali semantično oz. vsebinsko popolnoma drugačni odgovori od teh oddaljeni kar precej. Res je, v praksi so vgradnje predstavljene z več dimenzijami (pri GPT4 je to 1535 dimenzij) in ne dvema, kot je to za primer te vizualizacije. Posamezne dimenzije ni smiselno interpretirati. Vsekakor pa kombinacije vrednosti v različnih dimenzijah besedilu pripišejo semantični pomen – postopek kako pa to poteka, pa nam zaenkrat še ni popolnoma poznan.



Slika 2: prostor vgradenj, kjer so podobna besedila bližje skupaj.

V praksi te vgradnje niso v dveh ali treh dimenzijah, temveč v stotinah ali celo tisočih dimenzij. To je morda težko predstavljivo, vendar omogoča, da model predstavlja veliko število žetonov, od katerih ima vsaka svoj edinstven pomen. Ko jezikovni model prebere stavek, najprej vsak žeton pretvori v ustrezno vgradnjo – njegov edinstven prstni odtis. Ta postopek je kot prevajanje stavka iz človeškega jezika v jezik, ki ga model razume. Ko je stavek v tem *jeziku modela*, lahko model začne z zapleteno nalogo razumevanja stavka in ustvarjanja ustreznega rezultata.

Posebnost vgradenj je v tem, da vsebujejo več kot le odtis enega žetona. Vsebujejo tudi informacije o kontekstu žetona v okviru celotnega stavka. Ko velik jezikovni model obdeluje besedilo, nenehno posodablja vgradnje in jih spreminja, da bolje odražajo kontekst, v katerem je vsak žeton uporabljen. Na primer, v stavku »Zlomil si je prst« se vgradnja žetona »prst« posodobi tako, da odraža njegov pomen, povezan z delom roke, za razliko od »prst« v povedi »Prst je bila polna hranilnih snovi«, kjer ta pomeni geografski izraz za zgornjo plast zemeljske skorje.

Če povzamemo, vgradnje so način za predstavitev in razumevanje žetonov, saj vsebujejo informacije ne le o njihovi identiteti, temveč tudi o njihovih semantičnih pomenih in kontekstih, v katerih se uporabljajo. Ta numerična predstavitev je osnova za vse, kar model počne, od razumevanja vhodnega besedila do ustvarjanja kontekstualno ustreznih odgovorov.

4 Od vprašanja do odgovora

Proces, s katerim veliki jezikovni modeli, kot je ChatGPT, obdelujejo vprašanja in oblikujejo odgovore, predstavlja kompleksen sistem z več plastmi analize in sinteze. Ta postopek, čeprav na prvi pogled morda zapleten, je mogoče razčleniti v serijo strukturiranih korakov. V tem poglavju bomo podrobno preučili te korake in pojasnili, kako jezikovni model prehaja od prejema vprašanja do končnega odgovora.

Tokenizacija: Ko uporabnik postavi vprašanje ali pošlje sporočilo, ChatGPT v prvem koraku ta vnos razdeli na obvladljive dele, znane kot žetoni, o katerih smo že govorili.

Vstavljanje: Ko ima model svoje žetone, vsakega od njih pretvori v številčno obliko oz. vgradnjo o kateri je bilo več napisano v prejšnjem poglavju. Te vgradnje predstavljajo žetone na način, ki ga model lahko razume in z njim dela.

Razumevanje vprašanja: ChatGPT nato začne analizirati uporabnikovo vprašanje. Pri tem uporablja svojo arhitekturo transformerjev, vključno z mehanizmom pozornosti, da ugotovi, kako so žetoni (in njihove vstavitve) povezani drug z drugim. To je nekako tako, kot če bi poskušali razumeti zaplet zgodbe tako, da bi upoštevali, kako so liki povezani med seboj.

Začetek odgovora: Postopek generiranja se začne s posebnim žetonom, ki se pogosto imenuje žeton začetka zaporedja, ki je označba, ki nakazuje, da bo začel ustvarjati nov odziv. To je podobno kot velika tiskana črka na začetku stavka, ki naznanja začetek nove misli.

Napovedovanje naslednjega žetona: Po žetonu začetka odgovora model predvidi naslednji žeton. Tega ne stori v vakuumu. Namesto tega uporabi kontekst, ki ga zagotavlja uporabnikovo vprašanje, in vse žetone, ki jih je že ustvaril za odgovor. Model izračuna verjetnosti vseh možnih naslednjih žetonov in izbere enega naključnega, kjer pa je možnost za izbiro vsakega prilagojena na že obstoječi kontekst (dosedanja vprašanja in odgovore).

Na primer, če je uporabnik vprašal: »Kaj je prestolnica Francije?« in je model začel svoj odgovor z »Prestolnica«, je naslednji žeton lahko »Francije«. To je zato, ker se je model na podlagi učnih podatkov naučil, da je »preistolnica« pogosta beseda, ki sledi besedi »Francije« v kontekstu vprašanja o geografiji.

Ponavljjanje postopka: Model nato zaporedju doda nov žeton »je« in na podlagi posodobljenega konteksta predvidi naslednji žeton »Pariz«. Ob koncu odgovora model ustvari žeton za konec zaporedja oz. odgovora in s tem zaključi dopolnjevanje, ki se v našem primeru glasi »Prestolnica Francije je Pariz«. Tako se odgovori tvorijo vedno do zaključnega žetona, ki označuje zaključek odgovora ali pa do najdaljše možne meje odgovora. Namreč, trenutni veliki jezikovni modeli imajo omejitev koliko žetonov lahko prejmejo in napovedo (pri GPT-4 je to okoli 32 ali 64 tisoč žetonov – odvisno od verzije).

Obravnava negotovosti: Včasih se lahko zgodi, da je model negotov glede naslednjega žetona, saj ima več žetonov podobno verjetnost. V takih primerih model uporabi strategijo, imenovano *temperatura*, ki v bistvu določa, koliko tveganja je model pripravljen sprejeti. Visoka temperatura pomeni, da je model bolj pripravljen tvegati in izbrati manj verjetne žetone, kar lahko privede do bolj ustvarjalnih, vendar morda manj natančnih odzivov. Nasprotno pa nizka temperatura pomeni, da model raje igra na varnem in se drži najbolj verjetnih žetonov in s tem tvori ponovljive odgovore, ki so bolj v skladu z besedilom, ki ga je model videl tekom svojega učenja. Posledica manjše temperature pa je manjša kreativnost in večja verjetnost ponavljanja že videnega.

Dekodiranje odziva: Ko je odziv v celoti ustvarjen, je še vedno v obliki žetonov. Zadnji korak je dekodiranje teh žetonov nazaj v človeku berljivo besedilo. To je kot sestavljanje sestavljanke, pri čemer se posamezni žetoni spremenijo nazaj v koherentno in smiselno besedilo.

Ustvarjanje odgovora je torej občutljiv ples med uporabo naučenih vzorcev iz že prej videni (učnih) podatkov, razlago konteksta, ki ga sestavlja uporabnikov vnos z dosedanjimi odgovori, in uporabo statističnih verjetnosti, da se v vsakem koraku pripravi najboljša možna napoved. Končni cilj takih modelov je ustvariti odgovor, ki ni le slovnično pravilen, temveč tudi kontekstualno in semantično ustrezen, zanimiv in smiseln za uporabnika. Ta cilj je včasih dosežen, drugič pa spet ne. V naslednji sekciji sledi pregled kakšni so lahko rezultati opisanega procesa odgovarjanja.

4.1 Kakšni pa so odgovori?

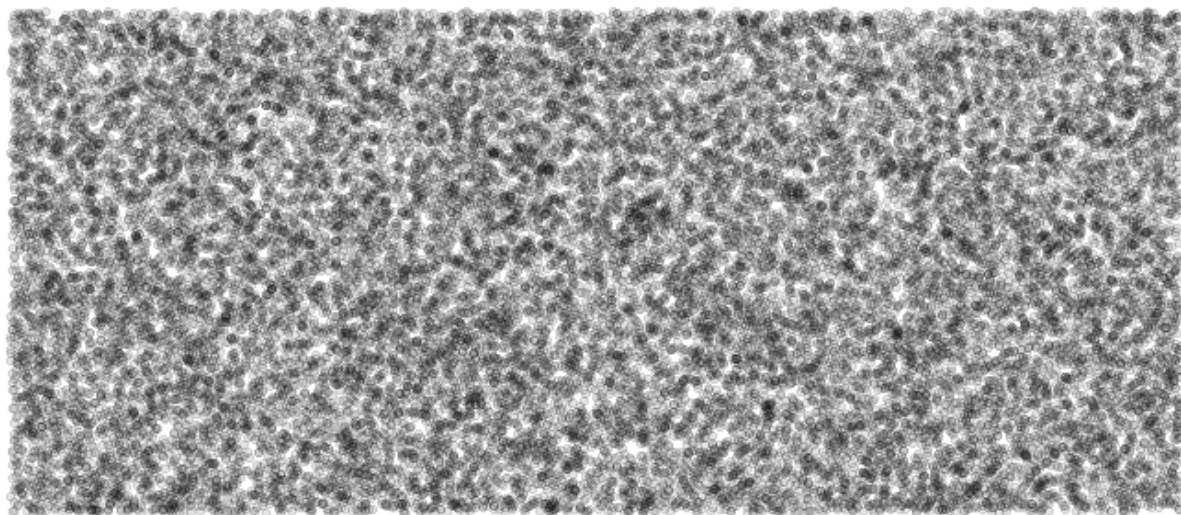
4.1.1 Nesmiselni odgovori

Veliki jezikovni modeli imajo pri odgovarjanju na določeno poizvedbo na voljo številne možne odgovore. Te odgovore oblikujejo na podlagi verjetnosti, ki so dodeljene vsakemu možnemu žetonu v njihovem besednjaku. Teoretično to pomeni, da lahko jezikovni model izbere kateri koli žeton, tudi najmanj verjeten, in tako ustvari odgovor, ki je lahko nesmiseln. Če vizualiziramo vse možne odgovore v prostoru, dobimo sliko 3. Pozor, vse sledeče slike v resnici ne prikazujejo odgovor vseh možnih odgovorov, saj bi prikaz toliko označb naenkrat privedel do nerazpoznavne vizualizacije. Posledično je prikazan le prostor, kjer je hipotetično možnih 1.000.000 odgovorov.



Slika 3: Prostor odgovor z vsemi možnimi odgovori. Slika je v celoti črna, ker so prikazani vsi možni odgovori: nesmiselni, smiselni; kratki (enoznakovni), polno dolgi; nepravilni in pravilni (na vsa možna vprašanja).

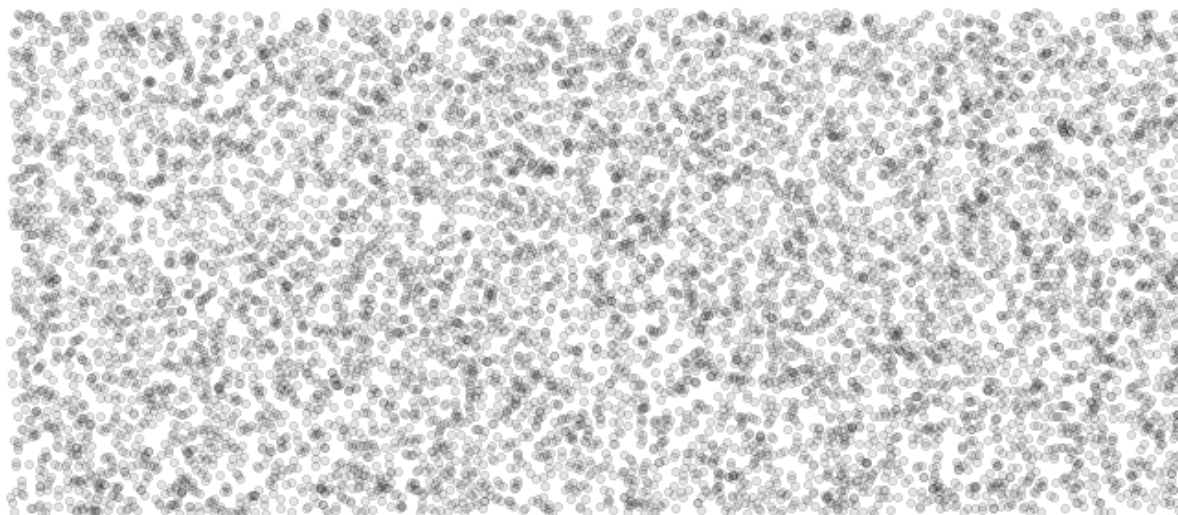
Drži, slika je popolnoma črna, ker vsaka pika predstavlja vse možne odgovore, tudi take, ki so nesmiselni, tudi kratke (enoznakovne) in dolge v vsej možno dolžini, tudi nepravilne in pravilne (za vsa možna vprašanja). Verjetnost, da pa model vrne nesmiselni odgovor, je zelo majhna, zato so takšni dogodki v običajnih okoliščinah praktično nemogoči. Kljub temu je tehnično izvedljivo, da model oblikuje odgovor, ki je sestavljen iz nerazumljivih zaporedij, kot je »njlkfdsjfklsa«. Če pa iz vizualizacije odstranimo vse nesmiselne odgovore, pa se znebimo velike večine vseh možnih kombinacij žetonov in nam tako ostane zelo majhen nabor (relativno glede na število vseh možnih odgovorov). Le smiselni odgovori so prikazani na sliki 4.



Slika 4: Prostor odgovor s smiselnimi odgovori. Ti odgovori še vedno lahko vsebujejo sintaktične (slovnične in tipkarske), semantične in logične napake ter neresnice.

4.1.2 **Smiselni odgovori s sintaktičnimi napakami**

Nekoliko večja verjetnost je, da veliki jezikovni modeli tvorijo smiselne odgovore, ki pa lahko še vedno vsebujejo sintaktične napake. Pri iskanju naslednjega najverjetnejšega žetona lahko jezikovni model občasno vključijo žetone, ki vodijo do sintaktičnih (slovničnih ali tipkarskih) napak. Nekatere od teh napak se lahko pojavijo naključno, vendar lahko glede na kontekstualno razumevanje jezikovni modeli pogosto takoj popravijo potek odgovora v naslednjih žetonih. Vendar pa začetna napaka ostane kot del odgovora, za razliko od človeške komunikacije, kjer lahko svoje napake popravimo za nazaj.



Slika 5: Prostor odgovor s smiselnimi odgovori brez sintaktičnih napak. Ti odgovori še vedno lahko vsebujejo semantične in logične napake ter neresnice.

Dodaten vir teh napak je mogoče razbrati iz podatkov za učenje takih velikih jezikovnih modelov. Če učni podatki vsebujejo tipkarske in slovnične napake, se lahko model nenamerno nauči teh vzorcev in jih ponovi v svojih rezultatih. Zato je pogostost pojavljanja napake v odzivih jezikovnega modela povezana s tem, kako pogosta je bila ta napaka v podatkih za usposabljanje.

Slika 5 prikazuje prostor odgovorov, kjer so nadaljnje odstranjeni še odgovori s sintaktičnimi napakami. Čeprav bi nam odgovori z malo in majhnimi sintaktičnimi napakami še vedno lahko prišli prav (saj so lahko pravilni), pa v namen nadaljnje razlage te odstranimo za namen bolj pregledne vizualizacije.

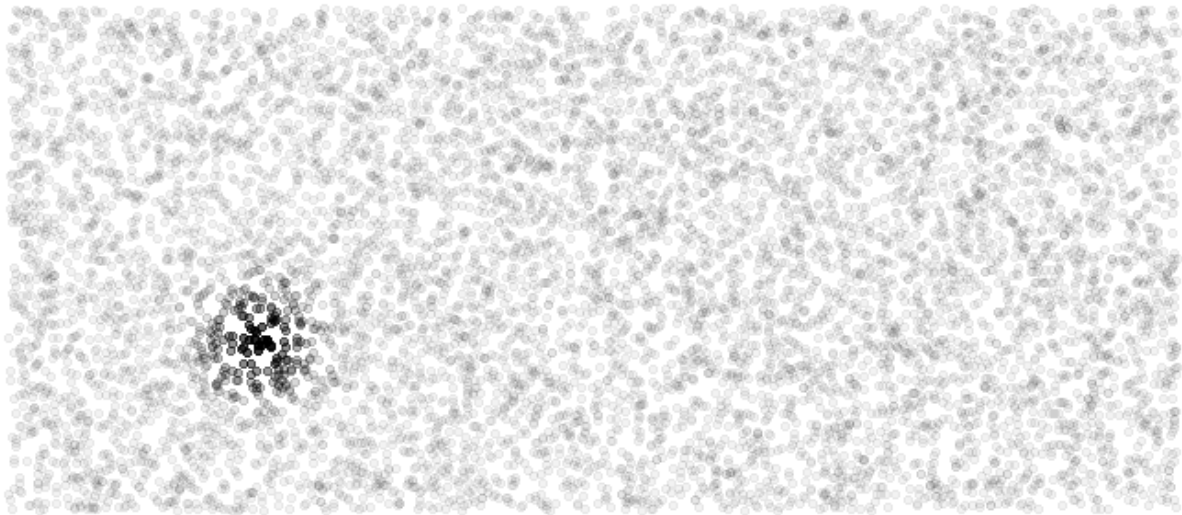
4.1.3 **Smiselni a napačni odgovori brez napak**

Veliki jezikovni modeli, ki so v večini primerov zelo natančni, lahko ustvarijo tudi razumne odgovore, ki pa so dejansko napačni. Do tega pojava lahko pride zaradi dveh ključnih razlogov.

Prvič, če učni podatki modela lahko vsebujejo nepravilne in neresnične informacije in ob učenju iz njih se model lahko nauči teh neresnic in jih ponovi v svojih odgovorih. Ta težava je še posebej izrazita, kadar se poizvedbe nanašajo na teme *v distribuciji* – to so teme, ki so dobro zastopane v učnih podatkih. Ker model generira svoje odgovore na podlagi vzorcev, ki se jih je naučil iz teh podatkov, se lahko napačne informacije, nehote predstavijo kot dejstva. Verjetnost, da se to zgodi, se povečuje s pogostostjo netočnih informacij v učnih podatkih, zlasti če so povezane z manj kontroverznimi temami (temami, katerih nasprotovanj je zelo malo v učnih podatkih).

Drugič, netočnosti se lahko pojavijo tudi pri obravnavi poizvedb *izven distribucije* ali poizvedb, ki se nanašajo na informacije, ki niso zajete v učnih podatkih. V teh primerih mora jezikovni model na podlagi naučenih vzorcev oblikovati utemeljena ugibanja, čeprav v svojem učnem korpusu nikoli ni videl pravega odgovora. Ta ugibanja so lahko včasih napačna, kar imenujemo *halucinacije*.

Sedaj pa slika 6 prikazuje vse možne odgovore s prilagojeno prosojnostjo glede na njihovo verjetnost, da bodo tvorjeni za eno fiktivno vprašanje (za en kontekst). Temnejši ko je odgovor, večja verjetnost je, da bo to končen odgovor. Če bi postavili popolnoma drugo vprašanje, bi gruča manj prosojnih (temnejših) oz. bolj verjetnih odgovorov bila popolnoma drugače.



Slika 6: Prostor odgovor s kjer so odgovori s prosojnostjo označeni glede na njihovo verjetnost, da bodo tvorjeni za podano fiktivno vprašanje. Manj prosojni imajo večjo verjetnost, da bodo tvorjeni.

Recimo, da smo jezikovni model prosili, naj nam tvori uvodno vabilo na našo konferenco. Že, če sami premislimo, bi to poleg dejansko uporabljenega stavka »Vabljeni na 26. konferenco Sodobne informacijske tehnologije in storitve« lahko ustvarili številne alternative, ki be imele enak semantični pomen. In prav te alternative bi bile temnejše na sliki 6. Nekaj takih, ki so semantično blizu (ne nujno identično) in bi bili temnejše obarvani na sliki je glede na ChatGPT sledečih:

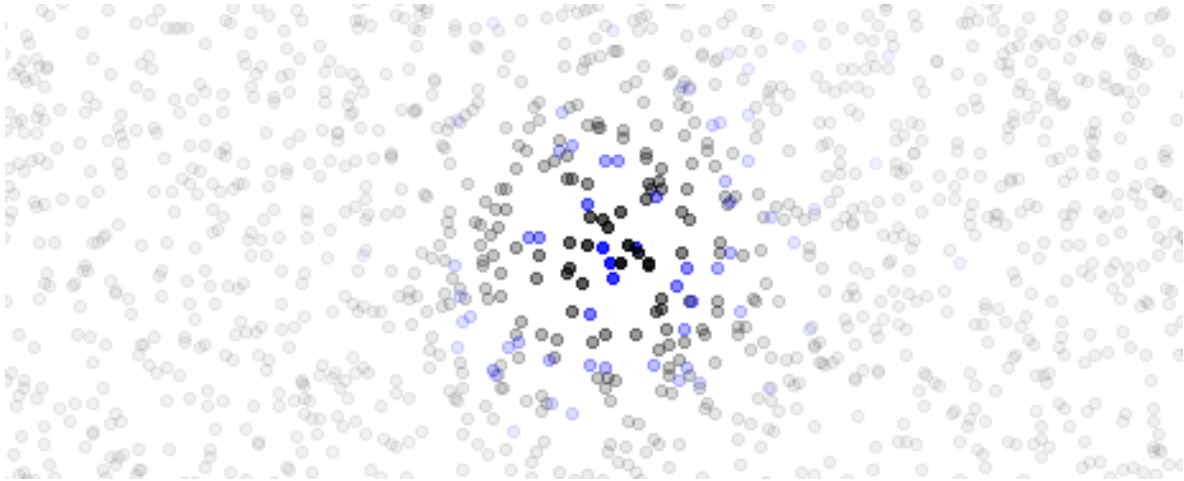
- *Dobrodošli na 26. konferenci Sodobne informacijske tehnologije in storitve.*
- *Pozdravljamo vas na 26. konferenci Sodobne informacijske tehnologije in storitve.*
- *Z veseljem vas vabimo na 26. konferenco o Sodobnih informacijskih tehnologijah in storitvah.*
- *Toplo vabljeni na konferenco Sodobne informacijske tehnologije in storitve, ki je že 26. po vrsti.*
- *Pridružite se nam na 26. konferenci na temo Sodobne informacijske tehnologije in storitve.*

4.1.4 Smiselni in pravilni odgovori brez napak

Kljub vsem do sedaj naštetim morebitnim pastem pa lahko veliki jezikovni modeli ustvarijo tudi povsem točne in razumne odgovore brez semantičnih ali sintaktičnih napak. Verjetnost takšnih popolnih odgovorov je večja pri poizvedbah v distribuciji, saj model pozna ustrezne vzorce v podatkih.

Tudi za poizvedbe zunaj distribucije lahko jezikovni modeli pripravijo pravilne in resnične odgovore. Čeprav so takšni rezultati precej odvisni od naključja, ima ključno vlogo sposobnost modela, da izlušči in uporabi semantične in logične vzorce iz svojega učnih podatkov. Z uporabo teh naučenih vzorcev v novih situacijah lahko model ustvari nove in natančne odgovore, kar kaže na njegov močan potencial kot orodja za generiranje besedila.

Slika 7 prikazuje odgovore, kjer so pravilni obarvani modro, kjer pa še vedno ohranimo prosojnostjo, ki prikazuje verjetnost za tvorjen odgovor. Tokrat pa zaradi boljše preglednosti slika prikazuje le povečavo področja, kjer je večja verjetnost kreiranih odgovorov. Iz slike je možno razbrati, da niso prav vsi najverjetnejši odgovori tudi pravilni. Hkrati pa so nekateri pravilni odgovori tudi taki, katerih tvorjenje ima manjšo verjetnost.



Slika 7: Prostor odgovor s kjer so pravilni odgovori obarvani modro, s prosojnostjo pa se prikazuje verjetnost, da bo odgovor kreiran.

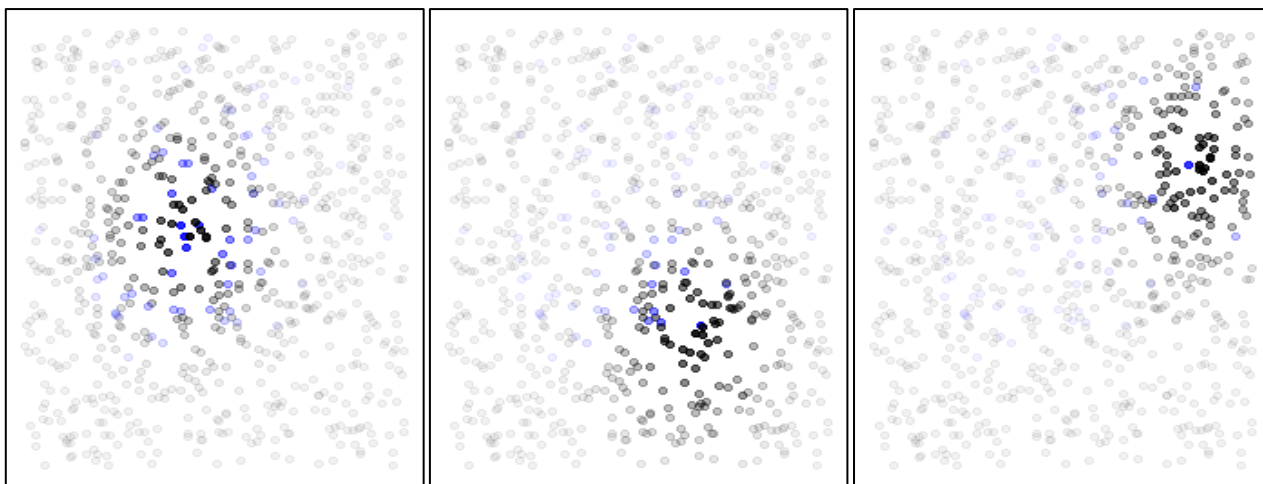
Pri postavljanju vprašanj imamo vsekakor cilj, da dobimo odgovor modela, ki bo pravilen (v vseh vidikih) oz. da vsaj povečamo možnost, a nam model vrne pravilen odgovor. Na to lahko vplivamo na dva načina: da prestavljamo center najbolj verjetnih odgovorov in da zožimo oz. zmanjšamo gručo najverjetnejših odgovorov.

4.2 Ovrednotenje pravilnosti odgovorov

Začnimo s *prestavljanjem centra najbolj verjetnih odgovorov*. Vidimo, da presek pravih in najbolj verjetnih odgovorov ni nujno popoln. Če za malenkost prilagodimo vprašanje (spremenimo kontekst), se lahko verjetnost (prosojnost) odgovorov drastično premakne, kljub temu, da pravilni odgovori (modro obarvani) ostanejo enaki. Naš cilj je vsekakor tvoriti taka vprašanja, da bi bili pravilni in najbolj verjetni odgovori v popolnem preseku. To je prikazano na sliki 8.

Če imamo znanje, da sami ocenimo pravilnost podanega odgovora, lahko pridemo vsaj do zaključka, če je vprašanje potrebno popraviti ali ne. Če se odločimo za popravek vprašanja, pa je izziv, da ob tvorjenju vprašanj, takih grafov ne vidimo (saj tudi ni računskih virov, da bi veliki jezikovni modeli izračunali verjetnost vseh možnih odgovorov), zato težko popravljamo vprašanja s ciljem popolnega preseka. Ponovno, če imamo znanje, je dobra praksa, da z izpostavljanjem napak prejšnjih odgovorov, z dodatnimi vprašanji ali prestrukturiranjem izvirnega vprašanja pridemo do pravih odgovorov.

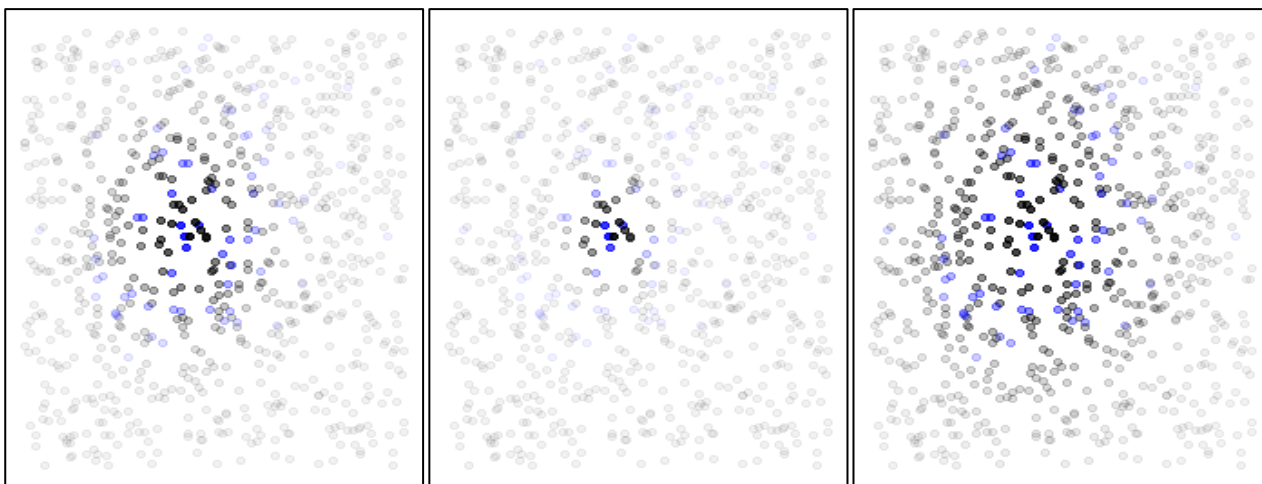
Če pa znanja za ocenitev pravilnosti odgovora ne posedujemo, pa je nemogoče identificirati halucinacije in nepravilnosti v odgovorih. Prav to je ena izmed pomanjkljivostih trenutnih jezikovnih modelov – da je njihova uporabnost zelo omejena, če jih uporabljamo za delo, katero ne vemo oceniti, če je pravilno ali ne. V primeru, da naloga ni občutljive narave (kreativno tvorjenje besedila ali kode) in katerih neprimerna uporaba ne bo povzročila resnih posledic, s tem večjega izziva ni. Če pa iščemo odgovore, ki bodo igrali pomembno vlogo v pomembnih odločitvah, ki imajo lahko resne posledice, pa je spraševanje o temah, ki jih sami ne obvladamo izredno zaskrbljujoče.



Slika 8: Prikaz različno oblikovanih ampak semantično enakih vprašanja.

4.3 Postavljanje primernih vprašanj

Naslovimo še z drugim načinom vplivanja na verjetnost, da dobimo pravilni odgovor: omejitvijo nabora najverjetnejših odgovorov. Če postavljamo vprašanja, ki sličejo na kratke in sintaktično nepravilne iskalne nize, ki smo se jih desetletja učili in uporabljali v iskalnikih (Google, Bing, DuckDuckGo...), bo nabor bolj verjetnih odgovorov mnogo večji. Večji ko je nabor najverjetnejših odgovor, večjo vlogo igra naključje kateri odgovor bomo dobili. Hkrati pa večji nabor najverjetnejših odgovorov razširi temnejšo gručo in s tem potencialno poveča razliko med presekom in ostankom z naborom pravih odgovorov.



Slika 9: Prikaz preoblikovanja vprašanja, ki zoži ali poveča nabor najverjetnejših odgovorov in s tem spremeni presek z naborom pravih odgovorov.

Po drugi strani pa tem, če bolj vprašanja napolnimo s podrobnostmi in navodili za odgovor, manjši bo obseg najbolj verjetnih odgovorov. Tako zmanjšamo vlogo naključja in zmanjšamo (v idealnosti celo izničimo) razliko med presekom z pravih odgovori. Preoblikovanje vprašanja, ki zoži ali poveča nabor najverjetnejših odgovorov je prikazan na sliki 9.

Za zoženje nabora možnih odgovorov uporabimo enake tehnike, kot bi jih uporabili z delom z ljudmi pri iskanju odgovorov. Praksa kaže, da se zelo dobro obnesejo vprašanja, kjer je hkrati podanih več podvprašanj, ki peljejo do končnega vprašanja. Na tako strukturo bo jezikovni model sproti odgovarjal najprej na podvprašanja in ko pride do dela, kjer odgovarja na krovno vprašanje, bodo del njegovega konteksta že njegovi odgovori na podvprašanja. Še vedno obstaja verjetnost, da bo tvoril napačen odgovor, ampak bo ta manjša, saj bo verjetno že ob podanem podvprašanju za to manjša.

Druga tehnika je, da vsilimo iskanje odgovora po korakih (angl. *chain-of-thought*). Pri takem načinu preprečimo »na blef« odgovarjanje in vsilimo, da model najprej »premisli« o vprašanju in razloži postopek kako je prišel do odgovora. Ta tehnika deluje zaradi enakega razloga kot prejšnja, verjetnost, da bodo celoten postopek, razmišljanje in končen odgovor modela napačno je manjša kot pa je verjetnost, da bo napačen le kratek odgovor. Tipično vprašanje, ki skorajda obvezno potrebuje tak pristop je reševanje matematičnih enačb. Tako ljudje, kot tudi jezikovni modeli potrebujejo vmesne izračune, da pridemo do rešitve zapletenih matematičnih nalog.

Naslednja tehnika je ponovno povzeta po delu z ljudmi – to je tehnika, ko poleg vprašanja modelu priložimo še en (*single-shot*) ali več (*multi-shot*) primernih ali pravilnih odgovorov. Pri takem postopku bi tako ljudje, kot sam model, skopiral postopek odgovarjanja, le priredili bi ključen del odgovora, da bi ta bil smiselna za podano vprašanje. Dober primer kjer to deluje odlično je pisanje programske kode z regexom. Če modelu poleg prošnje za kodo podamo tudi kako bi naj z gledali pravilni ekstrahirani nizi znakov (tem lahko podamo tudi nepravilne), je večja verjetnost, da bo vrnjena programska koda pravilna. Ta tehnika je tudi zelo primerna, če želimo odgovore točno v določeni obliki (npr. formalno pismo, v francoskem jeziku, v JSON ali XML formatu...).

5 Zaključek

Ta prispevek je poskušal pojasniti nekaj osnovnih delovnih načel velikih jezikovnih modelov, ne da bi se potopil v globoko teoretično razpravo ali se zapletal v filozofske dileme, kot je problem kitajske sobe. Pregledali smo štiri scenarije, kako veliki jezikovni modeli delujejo v odgovarjanju na vprašanja, ki nudijo vpogled v različne ravni uspešnosti in točnosti teh modelov. Upamo, da bo ta jasen in "nazoren" pregled pripomogel k zagonu produktivnih debat, brez zmede in neupravičenih pričakovanj, in s tem prispeval k bolj uravnoteženemu in razumnemu pristopu k tej hitro rastoči in vznemirljivi tehnologiji.

Literatura

- [1] VASWANI Ashish, SHAZEER Noam, PARMAR Niki, USZKOREIT Jakob, JONES Llion, GOMEZ Aidan N., KAISER Łukasz, in POLOSUKHIN Illia »Attention is all you need«, *Advances in neural information processing systems*, številka 30, leto 2017.
- [2] FLORIDI Luciano in CHIRIATTI Massimo "GPT-3: Its nature, scope, limits, and consequences.", *Minds and Machines*, številka 30, leto 2020, 681-694.
- [3] LIN Tianyang, WANG Yuxin, LIU Xiangyang in QIU Xipeng "A survey of transformers." *AI Open*, leto 2022.
- [4] LIU Xiao, ZHENG Yanan, DU Zhengxiao, DING Ming, QIAN Yujie, YANG Zhilin in TANG Jie "GPT understands, too." *arXiv preprint arXiv:2103.10385*, leto 2021.
- [5] ETHAYARAJH Kawin. "How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings." *arXiv preprint arXiv:1909.00512*, leto 2019.
- [6] DIS Van, AM Eva, BOLLEN Johan, ZUIDEMA Willem, VAN ROOIJ Robert in BOCKTING Claudi L. "ChatGPT: five priorities for research." *Nature* 614, številka 7947, leto 2023, 224-226.