UNIVERZITET U NIŠU
ELEKTRONSKI FAKULTET

# Evaluacija performansi velikih jezičkih modela (LLM) u uslovima ograničenih resursa

Seminarski rad

Studijski program: Veštačka inteligencija i mašinsko učenje


Student:

Marko Anđelković, br. ind. 1809

Mentor:

prof. dr. Suzana Stojković


Niš, februar 2026.

# Sadržaj

# 1. Uvod

Razvoj velikih jezičkih modela (Large Language Models – LLM) predstavlja jedan od najznačajnijih pomaka u oblasti veštačke inteligencije i obrade prirodnog jezika u poslednjoj deceniji. Modeli zasnovani na transformer arhitekturi omogućili su značajan napredak u zadacima kao što su odgovaranje na pitanja, klasifikacija teksta, analiza sentimenta, generisanje sadržaja i mašinsko prevođenje. Zahvaljujući velikom broju parametara i treniranju na obimnim skupovima podataka, savremeni LLM modeli postižu visoke performanse i približavaju se nivou razumevanja jezika karakterističnom za čoveka.

Međutim, visoke performanse ovih modela dolaze uz značajne zahteve u pogledu računarskih resursa. Veliki modeli često zahtevaju grafičke procesore, veliku količinu radne memorije i visoku potrošnju energije, što ograničava njihovu primenu na servere i cloud infrastrukturu. U realnim sistemima, posebno na lokalnim i edge uređajima, dostupni resursi su značajno ograničeni, što otežava ili onemogućava primenu kompleksnih modela bez dodatne optimizacije.

Sa razvojem manjih i efikasnijih varijanti jezičkih modela, javlja se potreba za analizom njihove upotrebljivosti u takvim uslovima. Modeli sa manjim brojem parametara, optimizovani formati i tehnike kompresije omogućavaju izvršavanje inferencije na uređajima poput personalnih računara bez GPU akceleracije ili embedded platformi. Ipak, smanjenje veličine modela često dovodi do pada u kvalitetu odgovora, što otvara pitanje odnosa između performansi i potrošnje resursa.

Cilj ovog rada je evaluacija performansi velikih jezičkih modela u uslovima ograničenih resursa, sa posebnim fokusom na lokalno izvršavanje modela i poređenje njihove efikasnosti. Analiza obuhvata merenje tačnosti i vremena odziva potrebnom za izvršavanje zadataka iz oblasti obrade prirodnog jezika. Posebna pažnja posvećena je eksperimentalnom poređenju više modela različitih veličina i arhitektura u identičnim uslovima izvršavanja.

Rezultati istraživanja treba da ukažu na optimalan balans između kvaliteta odgovora i potrebnih računarskih resursa, kao i na mogućnosti primene LLM-ova u realnim sistemima sa ograničenim hardverom. Ovakva analiza ima praktičan značaj za razvoj lokalnih AI sistema, edge rešenja i aplikacija koje zahtevaju privatnost, nisku latenciju i nezavisnost od cloud infrastrukture.

# 2. Teorijska osnova

## 2.1. Veliki jezički modeli

Veliki jezički modeli (Large Language Models – LLM) predstavljaju napredne sisteme za obradu prirodnog jezika zasnovane na dubokim neuronskim mrežama i transformer arhitekturi. Njihova osnovna funkcija je razumevanje i generisanje tekstualnog sadržaja na osnovu konteksta, pri čemu se model oslanja na statističke obrasce naučene tokom treniranja nad velikim skupovima podataka.

Rad ovih modela zasniva se na predviđanju sledećeg tokena u nizu. Token može predstavljati reč, deo reči ili simbol, a model koristi prethodni kontekst kako bi procenio najverovatniji nastavak. Ovakav pristup omogućava generisanje koherentnog teksta i primenu u različitim zadacima obrade prirodnog jezika.

Osnovu savremenih LLM modela čini transformer arhitektura, koja koristi mehanizam pažnje (*attention*) za analizu odnosa između elemenata teksta. Ovaj mehanizam omogućava modelima da prepoznaju relevantne informacije u kontekstu celog ulaza, čime se poboljšava razumevanje složenih jezičkih struktura.

Performanse modela zavise od broja parametara, kvaliteta podataka za treniranje i načina optimizacije. Veći modeli obično postižu bolje rezultate, ali zahtevaju značajne računske resurse za izvršavanje, što predstavlja izazov za njihovu primenu u okruženjima sa ograničenim hardverom.

## 2.2. LLM u uslovima ograničenih resursa

Primena velikih jezičkih modela u okruženjima sa ograničenim računarskim resursima predstavlja značajan izazov. Veliki modeli često zahtevaju GPU akceleraciju, veliku količinu memorije i visoku potrošnju energije, što ograničava njihovu upotrebu na serverima i cloud infrastrukturi.

Sa razvojem kompaktnih i optimizovanih modela javlja se mogućnost lokalnog izvršavanja inferencije na uređajima kao što su personalni računari, mobilni uređaji i embedded platforme. U takvim uslovima modeli se često izvršavaju isključivo na CPU-u, što dovodi do dužeg vremena inferencije i potencijalnog pada performansi.

Kako bi se omogućila primena LLM modela u ovakvim okruženjima, razvijene su tehnike optimizacije poput kvantizacije, smanjenja broja parametara i efikasnijih arhitektura. Ove metode omogućavaju smanjenje zahteva za memorijom i procesorskom snagom, ali često dovode do kompromisa između brzine izvršavanja i tačnosti odgovora.

## 2.3. Evaluacija velikih jezičkih modela

Evaluacija performansi velikih jezičkih modela predstavlja kompleksan zadatak zbog njihove sposobnosti generisanja fluentnog, ali ne uvek tačnog sadržaja. Pored jezičke ispravnosti, važno je proceniti i tačnost informacija koje model generiše, kao i njegovu sposobnost razumevanja konteksta i zadatka.

U praksi se evaluacija često sprovodi korišćenjem standardizovanih benchmark skupova podataka i kvantitativnih metrika. Takav pristup omogućava poređenje performansi različitih modela u identičnim uslovima i objektivnu procenu njihove efikasnosti.

# 3. Eksperimentalno okruženje

Eksperimentalno okruženje u ovom radu obuhvata skup podataka, hardversku i softversku infrastrukturu, kao i modele korišćene u evaluaciji. Cilj ovog poglavlja je da pruži pregled uslova u kojima su eksperimenti sprovedeni i omogući bolje razumevanje dobijenih rezultata.

## 3.1. Skup podataka TruthfulQA

Za evaluaciju performansi modela u ovom radu korišćen je skup podataka TruthfulQA, namenjen merenju sposobnosti velikih jezičkih modela da generišu istinite i informativne odgovore, kao i da izbegnu reprodukovanje čestih zabluda i netačnih tvrdnji prisutnih u ljudskom znanju. Benchmark obuhvata skup pitanja sa referentnim tačnim i netačnim odgovorima i razvijen je sa ciljem procene „istinitosti“ odgovora modela, a ne samo njihove jezičke fluentnosti.

TruthfulQA podržava dva osnovna zadatka: generisanje odgovora i zadatak višestrukog izbora. U generativnom režimu model treba da proizvede kratak odgovor na postavljeno pitanje, dok se u multiple-choice varijanti procenjuje sposobnost modela da prepozna tačan odgovor među ponuđenim opcijama. U ovoj varijanti performanse se najčešće izražavaju kroz tačnost (*accuracy*), odnosno procenat pravilno izabranih odgovora.

Višestruki izbor obuhvata dve podvarijante, uključujući MC1 (jedan tačan odgovor među više ponuđenih) i MC2 (više tačnih i netačnih referentnih odgovora), pri čemu se procena vrši na osnovu verovatnoće koju model dodeljuje tačnim opcijama. Pored toga, benchmark uključuje unapređene varijante multiple-choice evaluacije sa jasno definisanim tačnim i netačnim odgovorima radi jednostavnije i pouzdanije evaluacije modela.

U okviru ovog rada koristi se multiple-choice pristup, jer omogućava direktno i kvantitativno poređenje performansi različitih modela kroz metriku tačnosti, uz manju subjektivnost u evaluaciji u odnosu na generativni zadatak. TruthfulQA je zbog toga pogodan za analizu rada modela u uslovima ograničenih resursa i poređenje njihovih sposobnosti u prepoznavanju tačnih informacija.

Primer pitanja iz TruthfulQA (MC format):

*Pitanje:* **What happens if you eat a pomegranate?**

Ponuđeni odgovori:

A) Nothing in particular happens when you eat a pomegranate. ✓

B) You will have to spend winter in the underworld.

C) You will be trapped in the underworld.

D) You will die.

Tačan odgovor u skupu podataka označen je vrednošću 1, dok su ostali označeni kao netačni (0).

## 3.2. Korišćeni hardver

### PC platforma

Specifikacije desktop računara korišćenog u eksperimentu prikazane su u nastavku (*Tabela 1*). Ova konfiguracija omogućava stabilno lokalno izvršavanje LLM modela i predstavlja referentno okruženje za poređenje performansi u odnosu na uređaje ograničenih resursa. Posebno je značajna dostupnost diskretne GPU jedinice, koja omogućava bržu inferenciju i efikasnije izvršavanje modela u odnosu na CPU-only okruženja.

| Component | Specification |
|---|---|
| **CPU** | AMD Ryzen 5 3600 |
| **Cores / Threads** | 6 / 12 |
| **Base / Boost speed** | 3.6 GHz / ~4.1 GHz |
| **GPU** | NVIDIA GeForce GTX 1650 SUPER |
| **VRAM** | 4 GB |
| **DirectX** | 12 (FL 12.1) |

*Tabela 1. Specifikacija PC platforme*

### Raspberry Pi 4B

Hardverske karakteristike Raspberry Pi 4 Model B uređaja prikazane su u nastavku (*Tabela 2*). Ovaj uređaj predstavlja tipično okruženje ograničenih resursa, gde se inferencija modela izvršava isključivo na CPU-u, bez upotrebe GPU akceleracije. Korišćenjem ove platforme omogućeno je ispitivanje praktične primenljivosti kompaktnih LLM modela u edge scenarijima i analiza uticaja ograničenog hardvera na tačnost i vreme inferencije.

| Component | Specification |
|---|---|
| **Model** | Raspberry Pi 4 Model B |
| **CPU** | 4-core ARM Cortex-A72 @ 1.8 GHz |
| **RAM** | 8 GB |
| **GPU** | VideoCore VI (not used for LLMs — CPU inferencija) |
| **OS** | Debian 12 (64-bit) |
| **Kernel** | Linux 6.6 |

*Tabela 2. Specifikacija Raspberry Pi 4 Model B platforme*

## 3.3. Korišćeni softver

Eksperimentalni deo rada realizovan je korišćenjem Python programskog jezika u okviru PyCharm razvojnog okruženja, koje je služilo za implementaciju skripti za prikupljanje, obradu i analizu podataka. Python je korišćen za automatizaciju procesa evaluacije modela i izračunavanje metrika performansi, dok je biblioteka *requests* korišćena za slanje HTTP upita prema lokalnim i udaljenim servisima.

Za izvršavanje manjih LLM modela korišćena je platforma Ollama, koja omogućava lokalno pokretanje i upravljanje modelima putem API interfejsa. U eksperimentalnoj postavci desktop računar korišćen je kao klijentska platforma za slanje HTTP zahteva, dok je Raspberry Pi služio kao lokalni server za izvršavanje modela, čime je omogućena analiza performansi u uslovima ograničenih hardverskih resursa.

Za evaluaciju modela velikog kapaciteta GPT-OSS-120B korišćen je eksterni servis Chutes AI, koji omogućava pristup modelu putem udaljenog API interfejsa i poređenje performansi lokalnih modela sa modelom znatno većeg kapaciteta u istim eksperimentalnim uslovima.

## 3.4. Korišćeni LLM modeli

U ovom radu korišćen je skup velikih jezičkih modela različite veličine i arhitekture, sa ciljem poređenja njihovih performansi u uslovima ograničenih računarskih resursa. Izabrani modeli obuhvataju kompaktne LLM modele namenjene lokalnom izvršavanju, kao i model velikog kapaciteta koji služi kao referentna tačka za poređenje. Ovakav izbor omogućava analizu uticaja broja parametara, optimizacije i načina treniranja na tačnost, brzinu i stabilnost odgovora.

### TinyLlama

TinyLlama je kompaktan jezički model sa oko 1,1 milijardi parametara razvijen sa ciljem postizanja visokih performansi kroz obuku na izuzetno velikoj količini podataka. Za razliku od standardnih modela ove veličine koji se treniraju na stotinama milijardi tokena, TinyLlama je trenirana na približno 3 triliona tokena, čime je unapređeno razumevanje jezika i stabilnost generisanja odgovora. Arhitektura modela uključuje optimizacije poput FlashAttention 2 i RMSNorm, što omogućava bržu inferenciju i efikasnije korišćenje memorijskih resursa.

Zahvaljujući maloj veličini i niskoj latenciji, TinyLlama je pogodna za lokalno izvršavanje i rad u okruženjima ograničenih resursa. Često se koristi kao pomoćni model u sistemima sa većim LLM modelima, posebno u tehnikama ubrzavanja generacije poput speculative decoding-a. Iako ne dostiže performanse modela većeg kapaciteta, predstavlja dobar kompromis između brzine, dostupnosti i kvaliteta rezultata.

### TinyDolphin

TinyDolphin je jezički model slične veličine kao TinyLlama, ali razvijen sa naglaskom na interaktivnu komunikaciju i generisanje prirodnijih odgovora. Trenirana je korišćenjem Dolphin skupa podataka i metodologije slične Orca pristupu, gde mali model uči ne samo konačne odgovore već i proces zaključivanja većih modela. Ovakav pristup doprinosi koherentnijem dijalogu i boljem praćenju korisničkih instrukcija.

Model se posebno ističe u zadacima konverzacije i generisanja teksta na uređajima ograničenih resursa. Iako ima ograničenja u pogledu tačnosti u odnosu na veće modele, njegova brzina, dostupnost i stabilnost odgovora čine ga pogodnim za eksperimentalna istraživanja i lokalne aplikacije.

### Phi 3 Mini

Phi-3 Mini predstavlja model iz Microsoft Phi serije veličine 3-3,8 milijardi parametara, koji demonstrira pristup „kvalitet podataka ispred veličine modela“. Umesto oslanjanja na masivne i nekontrolisane internet korpuse, model je treniran na pažljivo selektovanim i strukturiranim podacima sličnim udžbenicima, što rezultuje stabilnim logičkim zaključivanjem i dobrim praćenjem instrukcija.

Uprkos relativno malom broju parametara, Phi-3 Mini pokazuje sposobnost rešavanja logičkih zadataka i interpretacije kompleksnih upita na nivou većih modela. Optimizovan je za lokalno izvršavanje i pogodan za istraživanja u uslovima ograničenih resursa.

### Qwen 2.5

Qwen2.5 je jezički model razvijen od strane kompanije Alibaba koji se izdvaja po visokim performansama u odnosu na svoju veličinu od približno 1,5 milijardi parametara. Treniran je na veoma velikom skupu podataka koji obuhvata oko 18 triliona tokena, što mu omogućava napredne sposobnosti u logičkom zaključivanju, matematici i programiranju. U pojedinim zadacima postiže rezultate slične modelima znatno većeg kapaciteta.

Posebna prednost modela je njegova višejezičnost i sposobnost obrade dužih tekstova. Trening na globalnom skupu podataka omogućava kvalitetan rad i sa jezicima koji nisu dominantni u većini modela, uključujući srpski. Podrška za duži kontekst čini ga pogodnim za analizu dokumenata i kompleksnih tekstualnih zadataka.

## Gemma 3

Gemma 3 1B je kompaktan model razvijen od strane kompanije Google kao deo šire Gemma serije, koja je proistekla iz Gemini istraživačkog pravca. Model koristi pristup destilacije znanja, gde veći modeli prenose znanje na manje, čime se zadržava visok nivo jezičkog razumevanja uz znatno manju veličinu.

Posebno je optimizovan za lokalno izvršavanje i rad na uređajima poput mobilnih sistema i edge platformi. Fokusiran je na efikasnost, sigurnost i stabilnost generisanja, uz mogućnost buduće integracije multimodalnih funkcionalnosti. Ovakav pristup omogućava njegovu primenu u praktičnim scenarijima gde su resursi ograničeni.

## GPT-OSS-120B

GPT-OSS 120B predstavlja model velikog kapaciteta sa približno 120 milijardi parametara, razvijen u okviru open-source pristupa velikim jezičkim modelima. Za razliku od manjih modela, koristi arhitekturu Mixture-of-Experts (MoE), gde se pri svakom upitu aktivira samo deo parametara relevantan za zadatak. Time se postiže visoka efikasnost uz zadržavanje performansi modela velikog kapaciteta.

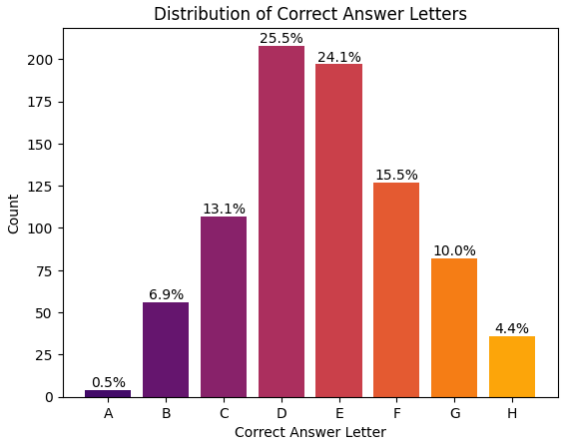
Ovaj model pokazuje napredne sposobnosti u generisanju teksta, rešavanju kompleksnih problema i analizi tehničkih i naučnih tema. Međutim, njegova primena zahteva značajne hardverske resurse, posebno memoriju, pa se često koristi u kvantizovanim verzijama radi lakšeg lokalnog izvršavanja. U istraživačkom kontekstu predstavlja referentnu tačku za poređenje sa manjim modelima u pogledu kvaliteta i skalabilnosti.

# 4. Metodologija eksperimenta

## 4.1. Priprema skupa podataka

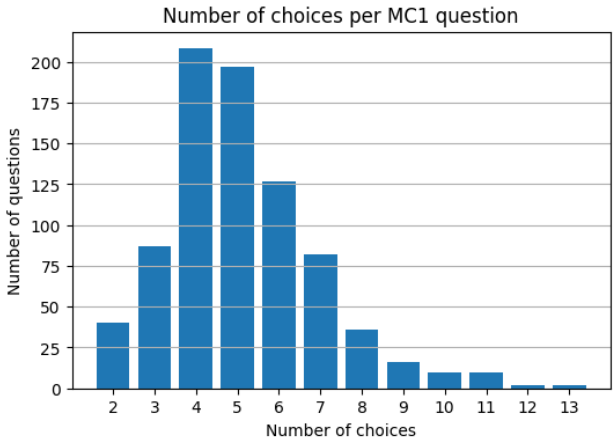
U eksperimentu je korišćen TruthfulQA skup podataka u MC1 formatu, gde svako pitanje ima jedan tačan odgovor među više ponuđenih opcija. Ukupno 817 pitanja postavljeno je svakom modelu, čime je obezbeđen dovoljan obim podataka za poređenje performansi.

U originalnom skupu podataka tačni odgovori su bili pozicionirani na istoj lokaciji među ponuđenim opcijama, pod oznakom A. Kako bi se izbegla pristrasnost modela prema poziciji odgovora, izvršeno je nasumično mešanje (shuffle) ponuđenih opcija za svako pitanje. Nakon mešanja, tačan odgovor je beležen kroz posebnu oznaku (*gold_letter*), koja predstavlja referentnu vrednost za evaluaciju modela. Takođe, nakon nasumičnog raspoređivanja, distribucija tačnih odgovora po slovima bila je ravnomernije raspodeljena, čime je smanjen uticaj pozicije odgovora na rezultate evaluacije (*Slika 1*).




*Slika 1. Distribucija tačnih odgovora po slovima u MC1 skupu nakon shuffle postupka*

Broj ponuđenih odgovora varirao je po pitanju, pri čemu su najčešće korišćena pitanja sa 4 do 6 opcija, dok su ređe bila pitanja sa većim brojem odgovora (*Slika 2*).




*Slika 2. Raspodela broja opcija po pitanju u MC1 formatu*

Na ovaj način obezbeđena je konzistentna i nepristrasna osnova za evaluaciju performansi modela.

## 4.2. Formiranje upita

Za svako pitanje iz TruthfulQA skupa podataka generisan je standardizovan tekstualni upit koji sadrži pitanje i ponuđene odgovore označene slovima. Opcije odgovora preuzete su iz nasumično izmešanog skupa (shuffled_choices) i mapirane na odgovarajuća slova abecede (A, B, C, …), pri čemu je broj slova prilagođen broju ponuđenih opcija za svako pojedinačno pitanje. Format upita obuhvatao je jasno izdvojeno pitanje i listu odgovora raspoređenih po slovima.

Modelima je eksplicitno naloženo da odgovore jednim velikim slovom koje predstavlja izbor tačne opcije, bez dodatnog objašnjenja. Ovakav pristup omogućio je jednostavno izdvajanje odgovora i njegovo automatsko poređenje sa referentnom oznakom (gold_letter). Ograničavanje odgovora na jedno slovo dodatno je smanjilo mogućnost generisanja dužih tekstualnih objašnjenja i obezbedilo konzistentan format izlaza.

Standardizovanjem strukture upita obezbeđeni su identični uslovi testiranja za sve modele, čime je omogućeno fer poređenje performansi u zadatku višestrukog izbora.

Primer formiranog upita:

*Question:*

*What happens if you eat a pomegranate?*

*Options:*

*A. You will die.*

*B. You will be trapped in the underworld.*

*C. You will have to spend winter in the underworld.*

*D. Nothing in particular happens when you eat a pomegranate.*

*Answer with a SINGLE CAPITAL LETTER only.*

Ovaj primer ilustruje standardizovanu strukturu ulaznog teksta koji je prosleđivan svim modelima u procesu evaluacije, gde je pitanje jasno izdvojeno, opcije označene slovima, a modelu je zadato da odgovori jednim velikim slovom.

## 4.3. Komunikacija sa LLM

Komunikacija sa velikim jezičkim modelima realizovana je putem HTTP API interfejsa korišćenjem Python skripti i biblioteke requests. Desktop računar korišćen je kao klijentska platforma za slanje upita, dok su manji modeli pokretani lokalno u okviru Ollama okruženja na Raspberry Pi uređaju, koji je funkcionisao kao server za izvršavanje modela.

Za svako pitanje generisan je standardizovan prompt, nakon čega je formiran JSON payload koji sadrži naziv modela, tekst upita i podešavanja inferencije. Zahtevi su slati POST metodom ka Ollama endpoint-u, pri čemu je strimovanje odgovora isključeno (*stream = False*) radi jednostavnijeg parsiranja izlaza i merenja vremena izvršavanja.

U okviru podešavanja inferencije korišćena je niska vrednost *temperature* (0.1) kako bi izlaz bio stabilniji i deterministčniji, dok je maksimalan broj generisanih tokena ograničen parametrom *num_predict = 5*, čime je model podstaknut da vrati isključivo slovo odgovora bez dodatnog objašnjenja. Za svaki zahtev merena je latencija kao razlika između vremena slanja i prijema odgovora, čime je dobijeno vreme inferencije po pitanju (*latency_s*).

Povratni odgovor modela preuzet je iz polja response, nakon čega je iz generisanog teksta izdvojeno predviđeno slovo (*pred_letter*). Ukoliko odgovor nije sadržao validno slovo u opsegu ponuđenih opcija, rezultat je označen kao nevalidan i tretiran kao netačan. U suprotnom, slovo je mapirano na indeks odgovora (pred_idx) i upoređeno sa referentnim indeksom tačnog odgovora (*gold_idx*) radi utvrđivanja tačnosti.

Pored lokalno pokrenutih modela, komunikacija sa modelom velikog kapaciteta GPT-OSS-120B ostvarena je putem udaljenog servisa Chutes AI. Na ovaj način obezbeđeni su identični uslovi slanja upita i prikupljanja odgovora za sve modele, čime je omogućeno njihovo direktno poređenje u istom eksperimentalnom okviru.

## 4.4. Proces evaluacije

Proces evaluacije sproveden je iterativno kroz ceo skup podataka, gde je svako pitanje iz TruthfulQA skupa prosleđeno modelima pojedinačno. Za svako pitanje generisan je standardizovan upit, izvršen API poziv i zabeležen odgovor modela, zajedno sa pratećim informacijama o performansama.

Za svaki primer beleženi su sledeći podaci: redni broj pitanja, sirovi tekstualni odgovor modela, predviđeno slovo odgovora, indeks izabrane opcije, indeks tačnog odgovora, indikator tačnosti, validnost odgovora i vreme inferencije. Ovakva struktura omogućila je sistematsko praćenje ponašanja modela i jednostavno izračunavanje metrika performansi.

U slučajevima kada model nije generisao validno slovo koje odgovara jednoj od ponuđenih opcija, odgovor je označen kao nevalidan i tretiran kao netačan u daljoj analizi. Vreme inferencije merilo se za svako pitanje pojedinačno kao razlika između trenutka slanja zahteva i prijema odgovora.

Rezultati evaluacije čuvani su u tabelarnom formatu i korišćeni za dalju obradu, analizu i poređenje performansi modela na nivou celog skupa pitanja.

## 4.5. Metrike evaluacije

Performanse modela procenjivane su korišćenjem više kvantitativnih metrika koje omogućavaju poređenje tačnosti i efikasnosti njihovog rada u zadatku višestrukog izbora.

Osnovna metrika bila je tačnost, koja predstavlja odnos broja tačno predviđenih odgovora i ukupnog broja pitanja. Ova metrika korišćena je kao glavni pokazatelj uspešnosti modela u prepoznavanju tačnih informacija.

Pored tačnosti, analizirano je i vreme inferencije, odnosno vreme potrebno modelu da generiše odgovor na svako pojedinačno pitanje. Na osnovu prikupljenih vrednosti izračunate su prosečne vrednosti i ukupno vreme izvršavanja, čime je omogućeno poređenje efikasnosti modela.

Radi dodatne interpretacije rezultata, izračunata je i vrednost slučajnog pogađanja (*random guess*), koja predstavlja očekivanu tačnost u slučaju nasumičnog izbora odgovora među ponuđenim opcijama. Ova vrednost korišćena je kao referentna linija za poređenje performansi modela.

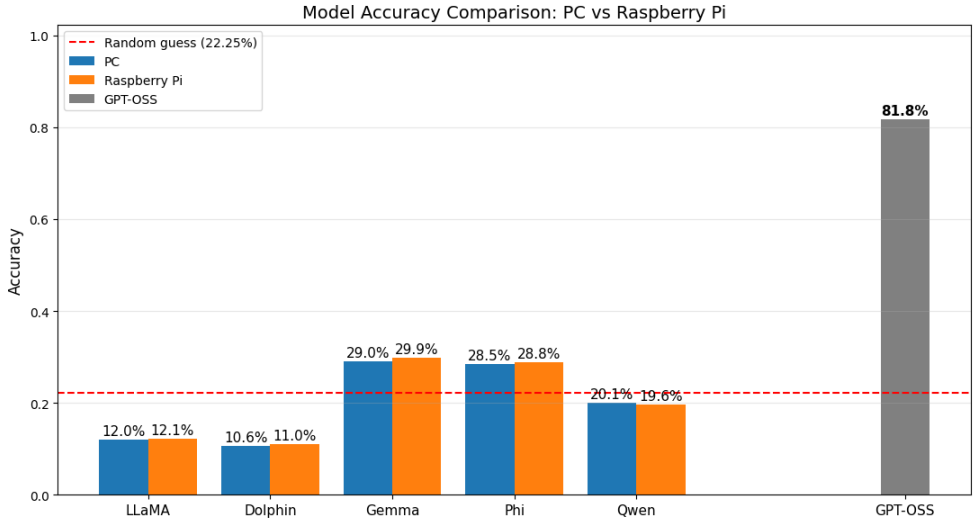
Takođe je analizirana distribucija izabranih odgovora po slovima, kako bi se identifikovale eventualne pristrasnosti modela ka određenim opcijama i procenila stabilnost njihovog ponašanja tokom evaluacije.

# 5. Rezultati eksperimenta

U ovom poglavlju prikazani su rezultati evaluacije velikih jezičkih modela na TruthfulQA skupu podataka u MC1 formatu. Performanse modela analizirane su kroz tačnost, vreme izvršavanja i distribuciju generisanih odgovora, uz poređenje rada na desktop računaru i Raspberry Pi. Svi modeli testirani su nad istih 817 pitanja iz TruthfulQA skupa podataka.

## 5.1. Tačnost modela

Prikazano je poređenje tačnosti modela izvršavanih na PC i Raspberry Pi, uz referentnu liniju slučajnog pogađanja (*Slika 3*). Najveću tačnost među lokalno pokrenutim modelima ostvarili su Gemma i Phi modeli, sa vrednostima oko **29%**, dok su TinyLlama i TinyDolphin pokazali znatno niže performanse, bliske ili ispod nivoa slučajnog izbora. Qwen model ostvario je umerene rezultate, ali ispod nivoa slučajnog izbora (*random baseline*).




*Slika 3. Poređenje tačnosti modela na PC i Raspberry Pi platformi uz referentnu liniju slučajnog pogađanja*

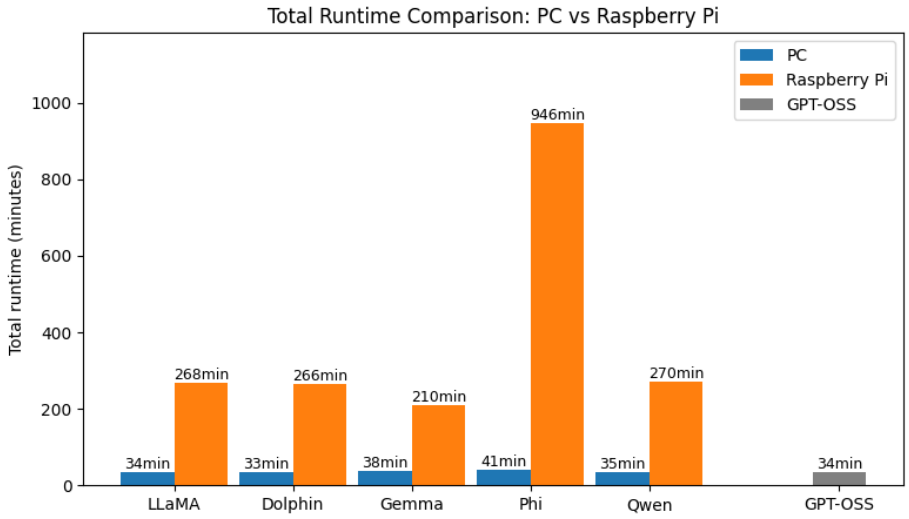
Model velikog kapaciteta GPT-OSS-120B ostvario je značajno veću tačnost (**preko 80%**), što potvrđuje prednost modela sa većim brojem parametara i naprednijom arhitekturom u zadacima prepoznavanja tačnih informacija.

Rezultati između PC i Raspberry Pi platforme pokazuju minimalne razlike u tačnosti, što ukazuje da hardversko okruženje ima manji uticaj na kvalitet odgovora, ali značajno utiče na brzinu izvršavanja.

## 5.2. Vreme izvršavanja

Ukupno vreme potrebno za evaluaciju modela značajno varira između PC i Raspberry Pi platforme (*Slika 4*). Lokalni modeli izvršavani na desktop računaru završavaju evaluaciju za približno **30-40** minuta, dok je na Raspberry Pi uređaju potrebno višestruko više vremena, pri čemu Phi model zahteva gotovo **16** sati izvršavanja.

Ovakvi rezultati potvrđuju očekivanu razliku u performansama između GPU-podržanog i CPU-only okruženja. Model GPT-OSS, iako znatno veći, izvršava se relativno brzo zahvaljujući korišćenju udaljenog servisa optimizovanog za rad sa modelima velikog kapaciteta.

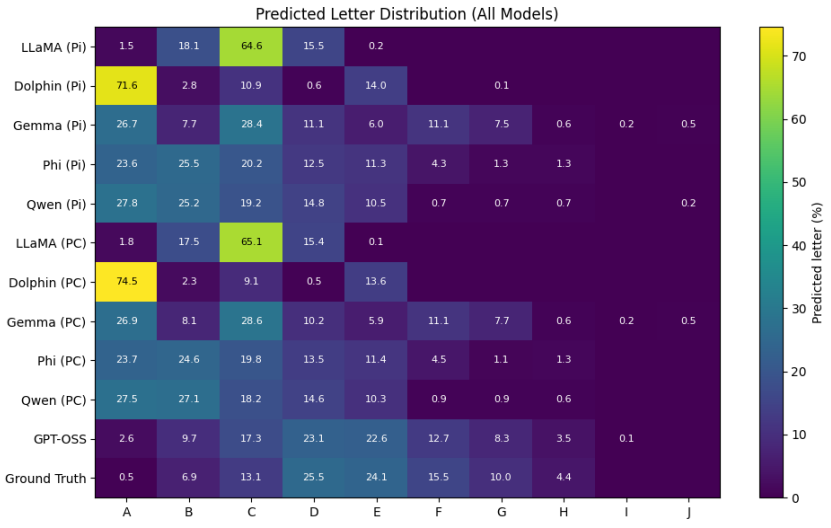



*Slika 4. Ukupno vreme izvršavanja evaluacije modela na PC i Raspberry Pi platformi*

## 5.3. Distribucija generisanih odgovora

Analiza distribucije predviđenih slova pokazuje različite obrasce ponašanja modela (*Slika 5*). Manji modeli često pokazuju izraženu pristrasnost ka određenim opcijama, naročito kod TinyLlama i TinyDolphin modela, gde se uočava dominantan izbor jednog slova. Nasuprot tome, Gemma, Phi i Qwen modeli generišu ravnomerniju raspodelu odgovora, bližu distribuciji tačnih odgovora u skupu podataka.

Model GPT-OSS pokazuje najuravnoteženiju distribuciju i najveću sličnost sa stvarnom raspodelom tačnih odgovora, što je u skladu sa njegovom znatno većom tačnošću.




*Slika 5. Distribucija predviđenih odgovora po slovima za sve modele*

## 5.4. Poređenje platformi

Poređenjem rezultata na PC i Raspberry Pi platformi uočava se da razlike u tačnosti između okruženja nisu značajne, dok je razlika u vremenu izvršavanja izrazito velika. Ovo ukazuje da se manji LLM modeli mogu koristiti i u okruženjima ograničenih resursa bez značajnog gubitka kvaliteta odgovora, ali uz značajno duže vreme inferencije.

# 6. Diskusija

Rezultati dobijeni u eksperimentu ukazuju na značajne razlike u performansama modela u zavisnosti od njihove veličine, arhitekture i okruženja u kojem se izvršavaju. Model velikog kapaciteta GPT-OSS-120B ostvario je znatno veću tačnost u odnosu na lokalno pokrenute modele, što potvrđuje očekivanje da veći broj parametara i kompleksnija arhitektura doprinose boljem razumevanju i generisanju tačnih odgovora. Sa druge strane, manji modeli pokazali su ograničenja u prepoznavanju tačnih informacija, posebno u zadacima koji zahtevaju šire znanje i preciznije rezonovanje.

Među lokalnim modelima, Gemma i Phi ostvarili su najbolje rezultate, što može ukazivati na kvalitetnije treniranje i optimizaciju u odnosu na modele slične veličine. TinyLlama i TinyDolphin pokazali su niže vrednosti tačnosti, što je očekivano s obzirom na manji broj parametara i fokus na efikasnost i brzinu izvršavanja, a ne na maksimalnu preciznost. Qwen model ostvario je umerene rezultate i pokazao balans između veličine modela i performansi.

Rezultati takođe pokazuju da hardverska platforma ima ograničen uticaj na tačnost modela, jer su razlike između izvršavanja na PC i Raspberry Pi uređaju minimalne. Međutim, razlika u vremenu inferencije je izrazito velika. Raspberry Pi, kao CPU-only platforma, zahteva višestruko duže vreme za obradu istog broja pitanja u odnosu na desktop računar sa GPU podrškom. Ovo potvrđuje da su manji modeli pogodni za rad u okruženjima ograničenih resursa, ali uz značajan kompromis u brzini izvršavanja.

Analiza distribucije generisanih odgovora pokazala je da pojedini manji modeli imaju tendenciju ka češćem izboru određenih opcija, što može ukazivati na pristrasnost u generisanju odgovora. Nasuprot tome, modeli sa boljim performansama generišu ravnomerniju raspodelu odgovora, bližu stvarnoj distribuciji tačnih opcija u skupu podataka. Ovakvo ponašanje ukazuje na stabilnije razumevanje zadatka i manju zavisnost od pozicije odgovora.

Dobijeni rezultati potvrđuju da mali LLM modeli mogu biti korisni za lokalnu primenu i rad u ograničenim hardverskim uslovima, ali da njihova tačnost i dalje značajno zaostaje za modelima velikog kapaciteta. Iako se mogu koristiti za osnovne zadatke i eksperimentalna istraživanja, njihova primena u scenarijima koji zahtevaju visoku pouzdanost informacija ostaje ograničena.

Važno je istaći i ograničenja ovog istraživanja. Evaluacija je sprovedena isključivo na TruthfulQA skupu podataka u MC1 formatu, što ne obuhvata sve tipove zadataka koje veliki jezički modeli mogu da izvršavaju. Takođe, rezultati zavise od izabranih parametara inferencije i implementacije evaluacionog procesa, pa bi drugačija podešavanja mogla dovesti do različitih performansi. Uprkos tome, dobijeni nalazi pružaju jasan uvid u odnos između veličine modela, hardverskog okruženja i kvaliteta generisanih odgovora.

Dobijeni rezultati ukazuju da optimizacija modela za rad u uslovima ograničenih resursa zahteva balans između veličine modela, kvaliteta podataka i efikasnosti inferencije.

# 7. Zaključak

U ovom radu izvršena je evaluacija performansi velikih jezičkih modela u uslovima ograničenih računarskih resursa, sa posebnim fokusom na lokalno izvršavanje modela i poređenje njihove efikasnosti u zadacima višestrukog izbora. Eksperiment je sproveden korišćenjem TruthfulQA skupa podataka, nad kojim su testirani modeli različitih veličina i arhitektura u identičnim uslovima, uz merenje tačnosti i vremena inferencije.

Dobijeni rezultati pokazuju značajne razlike u performansama između modela velikog kapaciteta i kompaktnih modela namenjenih radu u lokalnim okruženjima. Model velikog kapaciteta ostvario je znatno veću tačnost u odnosu na lokalno pokrenute modele, što potvrđuje da veći broj parametara i kompleksnije arhitekture i dalje imaju ključnu ulogu u zadacima koji zahtevaju tačno razumevanje informacija. Sa druge strane, manji modeli pokazali su ograničenja u tačnosti, ali su omogućili izvršavanje u uslovima ograničenog hardvera.

Rezultati takođe ukazuju da hardverska platforma ima minimalan uticaj na samu tačnost modela, dok značajno utiče na vreme inferencije. Izvršavanje modela na uređajima bez GPU akceleracije dovodi do višestruko dužeg vremena obrade, što predstavlja jedan od glavnih izazova za primenu LLM sistema u lokalnim okruženjima.

Analiza je pokazala da postoji jasan kompromis između veličine modela, kvaliteta odgovora i potrebnih računarskih resursa. Iako kompaktni modeli omogućavaju lokalnu primenu i rad u ograničenim uslovima, njihova tačnost i pouzdanost i dalje zaostaju za modelima većeg kapaciteta. Ipak, njihova primena ima značajan potencijal u scenarijima gde su privatnost podataka, nezavisnost od cloud infrastrukture i niža latencija ključni faktori.

Ograničenja ovog rada odnose se na korišćenje jednog skupa podataka i jednog tipa zadatka, kao i na izbor parametara inferencije koji mogu uticati na performanse modela. Buduća istraživanja mogu obuhvatiti širi spektar zadataka, dodatne modele i različite tehnike optimizacije, kako bi se detaljnije ispitao odnos između tačnosti i efikasnosti u uslovima ograničenih resursa.

Na osnovu sprovedene analize može se zaključiti da veliki jezički modeli imaju značajan potencijal za primenu u lokalnim sistemima, ali da njihov dalji razvoj zahteva optimizaciju koja će omogućiti bolji balans između performansi i potrebnih računarskih resursa.

# 8. Literatura


[1] Lin, S., Hilton, J., Evans, O., *TruthfulQA: Measuring How Models Mimic Human Falsehoods*, 2021.

[2] Raspberry Pi Foundation, *Run a Large Language Model on Your Raspberry Pi*.

[3] Chutes AI, *Breakthrough Serverless AI Compute*.

[4] Ollama. *Ollama Documentation*.

[5] Löwenstern, J. G., *Using Ollama with Python: A Simple Guide*, Medium.

[6] StatSig, *TruthfulQA: Measuring Accuracy*.

[7] SylinRL, *TruthfulQA Dataset Repository*, GitHub