Sustav zaštite kibernetičke sigurnosti pomoću maskiranja velikog jezičnog modela

Jurinčić, Dominik

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 447

SUSTAV ZAŠTITE KIBERNETIČKE SIGURNOSTI POMOĆU MASKIRANJA VELIKOG JEZIČNOG MODELA

Dominik Jurinčić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 447

SUSTAV ZAŠTITE KIBERNETIČKE SIGURNOSTI POMOĆU MASKIRANJA VELIKOG JEZIČNOG MODELA

Dominik Jurinčić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 447

Pristupnik:	**Dominik Jurinčić (0036521891)**
Studij:	Računarstvo
Profil:	Znanost o podacima
Mentor:	izv. prof. dr. sc. Marin Šilić

Zadatak: **Sustav zaštite kibernetičke sigurnosti pomoću maskiranja velikog jezičnog modela**

Opis zadatka:

U posljednje vrijeme sveprisutni su javno dostupni primjenski sustavi zasnovani na generativnoj umjetnoj inteligenciji kao što su različiti virtualni agenti zasnovani na velikim jezičnim modelima koji omogućuju automatizaciju poslovnih procesa i bržu dostupnost usluga za poslovne i krajnje korisnike.Poznavanje konkretne inačice velikog jezičnog modela koji je u pozadini virtualnog agenta omogućuje napadaču provođenje različitih kreativnih napada na pozadinski sustav što može ugroziti različite aspekte kibernetičke sigurnosti. U okviru diplomskog rada potrebno je proučiti i istražiti različite mogućnosti zaštite kibernetičke sigurnosti pozadinskih sustava zasnovane na prepoznavanju i maskiranju velikih jezičnih modela koji pogone javno izložene virtualne agente i usluge. Potrebno je prikupiti prikladne i javno dostupne skupove podataka za zadatak detekcije velikog jezičnog modela na osnovi interakcije korisnika s virtualnim agentom ili uslugom. Nadalje, potrebno je oblikovati i programski ostvariti sustav za detekciju velikog jezičnog modela primjenom različitih algoritama strojnog učenja. U sljedećem koraku potrebno je u sustav ugraditi maskiranje odgovora velikog jezičnog modela kako bi se potencijalnom napadaču onemogućilo prepoznavanje istog te pritom maskiranje ne smije utjecati na semantiku odgovara modela. Ispitati uspješnost ostvarenog sustava primjenom prikladno odabranih mjera te prikazati i opisati rezultate ispitivanja. Uz rad je potrebno predati i dokumentirati izvorni kod ostvarenog sustava, korištene skupove podataka te navesti korištenu literaturu.

Rok za predaju rada: 28. lipnja 2024.

Sadržaj

Uvod

U današnje vrijeme sve je više raširena uporaba velikih jezičnih modela (*eng. Large language models*). Tome u prilog govori i podatak da je popularni servis ChatGPT tvrtke OpenAI u prvih 60 dana od svojeg puštanja u pogon dosegao 100 milijuna korisnika [1]. Njihove mogućnosti obrade, strukturiranja, razumijevanja i generiranja prirodnog jezika čine ih korisnim, a sve češće i nezaobilaznim alatima za pomoć pri rješavanju svakodnevnih zadataka. Ovi modeli često se koriste u implementaciji javno dostupnih primjenskih virtualnih agenata za automatizaciju i ubrzavanja procesa kao što su pružanje korisničke podrške, prevođenje jezika i mnogi drugi. Njihove mogućnosti obrade prirodnog jezika također ih čine pogodim žrtvama raznih kibernetičkih napada. Zbog krajnje nepredvidivosti prirodnog jezika kojeg korisnici u interakciji s agentom ili sustavom zasnovanom na velikim jezičnim modelima mogu strukturirati na beskonačan broj načina, napadači su u stanju pažljivom izradom upita (*eng. prompt*) zaobići zaštitne sustave i natjerati model da im otkrije povjerljive informacije ili ga koristiti za svoje osobne potrebe koje nisu u skladu s namjenom agenta ili sustava. Sigurnost velikih jezičnih modela zbog toga je postala jedno od najvažnijih pitanja koje treba riješiti kako bi se osiguralo da njihova primjena ne dovede do neželjenih posljedica.

U sklopu ovog rada istražit ću velike jezične modele i aspekte njihove sigurnosti. U početku ću se osvrnuti na same modele i način na koji oni funkcioniraju. Zatim ću istražiti sigurnosne aspekte korištenja velikih jezičnih modela. Različiti veliki jezični modeli su podložni na različite strategije napada, stoga je jedan od najvažnijih podataka do kojih potencijalni napadač može doći točan model koji se koristi u implementaciji agenta ili sustava kojeg namjerava napasti. Kako bi se smanjila površina napada (*eng. attack surface*) potrebno je onemogućiti napadača u namjeri da sazna taj podatak. U skladu s tim, u nastavku rada ću istražiti mogućnosti prepoznavanja velikog jezičnog modela na temelju njegovih odgovora i mogućnosti maskiranja odgovora kako bi se to onemogućilo. Prepoznavanje modela ostvarit ću korištenjem odabranih modela strojnog učenja treniranih na skupu podataka u kojem ću prikupiti odgovore raznih velikih jezičnih modela. U konačnici ću pokušati iskoristiti maskiranje odgovora kako bih onemogućio najbolje modele dobivene u prethodnom koraku u prepoznavanju modela.

Cilj ovog rada je pružiti pregled područja velikih jezičnih modela i njihove sigurnosti istraživanjem mogućnosti detekcije modela na temelju njihovih odgovora i maskiranja odgovora sa svrhom onemogućavanja detekcije.

1 Jezični modeli

Obrada prirodnog jezika (*eng. natural language processing*) grana je računarske znanosti koja proučava mogućnosti računalnog razumijevanja jezika. Većina podataka, a time i znanja, koje ljudi proizvode sačuvana je u nestrukturiranom tekstualnom obliku. Ako toj činjenici pridodamo činjenicu da se količina podataka proizvedenih svake godine strelovito povećava, uviđamo da je mogućnost kvalitetne računalne obrade takvih tekstualnih podataka iznimno važan zadatak. Obrada takvih podataka vrlo je izazovna zato što je prirodni jezik nejasan, višeznačan i često se oslanja na opće znanje čitatelja za razumijevanje. Većina najvažnijih problema u domeni obrade prirodnog jezika, kao što su prevođenje, sažimanje teksta i odgovaranje na pitanja, problemi su pretvorbe niza u niz (*eng. sequence to sequence*). Kod takvih je problema cilj na temelju ulaznog niza riječi ili tokena odrediti valjani izlazni niz riječi ili tokena. Najbolji modeli u rješavanju ovih problema dugo vremena su bile razne verzije povratnih neuronskih mreža (*eng. recurrent neural networks*). Povratne neuronske mreže, za razliku od standardnih neuronskih mreža, imaju mogućnost rada s nizovima varijabilnih duljina, što je glavna karakteristika problema obrade prirodnog jezika [2]. Ovi modeli na ulazu dobivaju pojedinačne, slijedno poredane riječi ili tokene ulaznog niza, te za svaki ulaz računaju novu izlaznu vrijednost na temelju trenutnog ulaza i skrivenog vektora stanja u kojem su pohranjene informacije o prijašnjim dijelovima ulaznog niza. Formalno ovo možemo zapisati kao izraze (1) i (2):

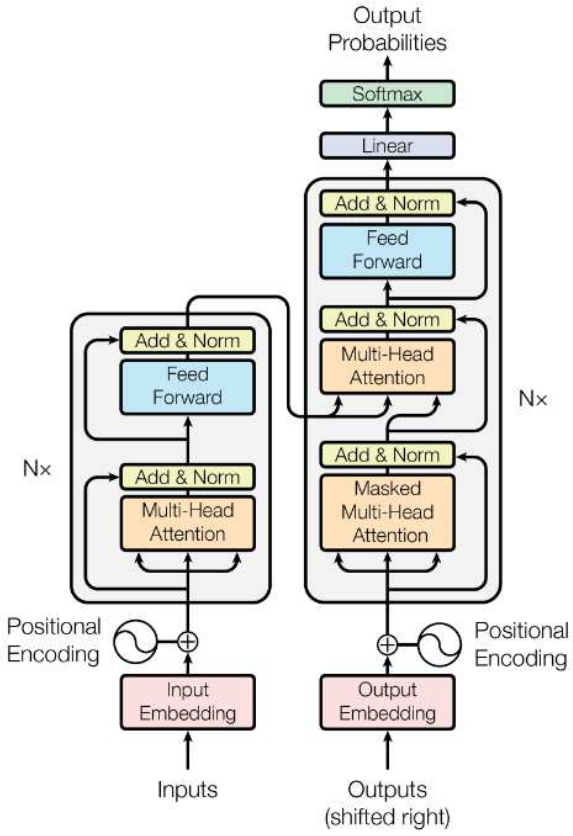
$$h_t = f_h(W_{hh} \cdot h_{t-1} + W_{xh} \cdot x_t), \quad (1)$$

$$y_t = f_y(W_{hy} \cdot h_t), \quad (2)$$

gdje h_t predstavlja skriveno stanje u trenutku t, odnosno nakon t ulaznih vrijednosti x, y_t predstavlja izlaznu vrijednost u trenutku t, funkcije f_h i f_y predstavljaju aktivacijske funkcije, a W_{hh}, W_{xh} i W_{hy} predstavljaju matrice težina. Iako su pružile veliki napredak u obradi nizova, povratne neuronske mreže imaju dva velika nedostatka. Prvi nedostatak je nemogućnost pamćenja dugoročnih povezanosti u ulaznom nizu, a drugi je pojava eksplodirajućih ili nestajućih gradijenata za vrijeme treniranja modela.

1.1 Transformeri

Arhitektura transformera [3] predstavlja sljedeći veliki korak naprijed u području obrade prirodnog jezika. Transformeri su zasnovani na mehanizmu pažnje (*eng. attention mechanism*) koji omogućava modeliranje međuovisnosti riječi u nizovima. Arhitektura transformera vidljiva je na slici (Slika 1.1Slika 1.1 Arhitektura transformera) preuzetoj iz rada [3].



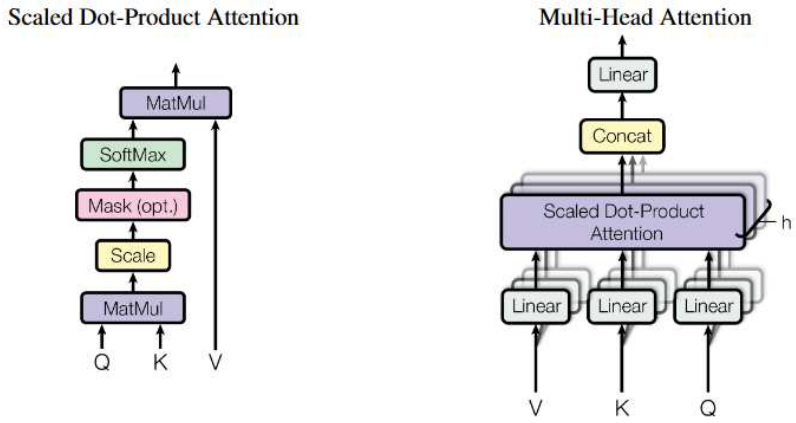
Slika 1.1 Arhitektura transformera

Transformeri su autoenkoderi koji se sastoje od kodera, koji je prikazan na lijevoj strani slike (Slika 1.1), i dekodera, koji je prikazan na desnoj strani slike (Slika 1.1). Razmotrimo kako transformeri funkcioniraju. Ulazni niz riječi se na početku zamijeni njihovim latentnim vektorskim reprezentacijama (*eng. embeddings*). Budući da transformeri obrađuju cijeli ulazni niz odjednom, a ne slijedno kao povratne neuronske mreže, potrebno je u reprezentacije ulaznih riječi dodati informacije o njihovom položaju u ulaznom nizu. To se postiže tako da se reprezentacije ulaznih riječi zbroje s reprezentacijama pozicija dobivenim formulama (3) i (4):

$$PE(pos, 2i+1) = cos\left(\frac{pos}{1000^{2i/d_{model}}}\right), \quad (3)$$

$$PE(pos, 2i) = sin\left(\frac{pos}{1000^{2i/d_{model}}}\right), \quad (4)$$

gdje se prvom formulom dobivaju reprezentacije neparnih pozicija, a drugom reprezentacije parnih pozicija. Zatim dolazimo do koderskog sloja koji se sastoji od sloja višestruke pažnje i potpuno povezanog unaprijednog sloja, između kojih su dodane rezidualne veze i slojevi normalizacije. Počnimo od sloja višestruke pažnje. Taj sloj za svaku od riječi ulaznog niza računa latentnu reprezentaciju čija se vrijednost temelji na naučenoj količini pažnje koju svaka riječ pridodaje ostalim riječima ulaznog niza. Prikaz sloja višestruke pažnje vidljiv je na slici (Slika 1.2) preuzetoj iz rada [3].



Slika 1.2 Sloj višestruke pažnje

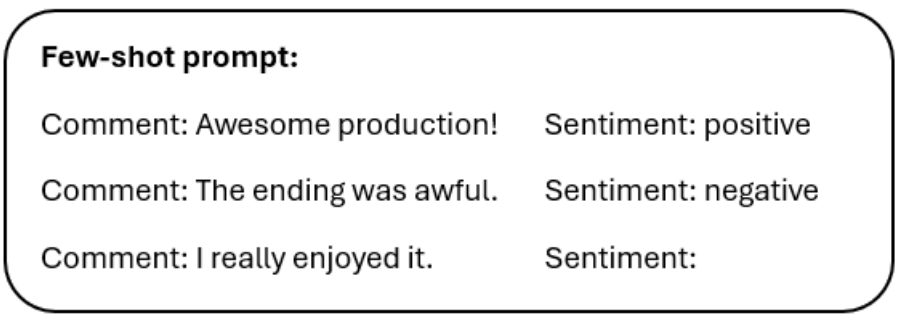
Za računanje reprezentacija sloj višestruke pažnje koristi vektorske reprezentacije upita (*eng. queries, Q*), ključeva (*eng. keys, K*) i vrijednosti (*eng. values, V*). Te vektorske reprezentacije dobivaju se propuštanjem reprezentacija ulaznih riječi kroz 3 različita potpuno povezana sloja. Matrice upita i ključeva se zatim matrično pomnože i skaliraju, a nad redovima rezultantne matrice, koji predstavljaju pozicije riječi u ulaznom nizu, primjenjuje se aktivacijska funkcija softmax. Rezultat tih operacija je matrica koja u svojim redovima sadrži vrijednosti koje predstavljaju relevantnost ostalih riječi u ulaznom nizu za riječ čija ulazna pozicija odgovora indeksu retka. Ta matrica se zatim matrično množi s matricom vrijednosti kako bi se dobile težinske sume vrijednosti koji predstavljaju reprezentacije pozicija ulaznog niza. Riječ „višestruka“ u nazivu sloja višestruke pažnje odnosi se na

činjenicu da se ovaj postupak izvodi paralelno s različitim parametrima u svakoj verziji izvođenja. Rezultati svih izvođenja se u konačnici konkateniraju i propuštaju kroz potpuno povezani sloj kako bi se dobile konačne reprezentacije. Tim reprezentacijama se zatim pribroje originalne reprezentacije riječi putem rezidualne veze i rezultat se normalizira. Normalizirani rezultat se propušta kroz potpuno povezani sloj, koji se sastoji od dva unaprijedna potpuno povezana sloja između kojih se reprezentacije propuštaju kroz aktivacijsku funkciju zglobnicu (*eng. rectified linear unit, ReLU*), nakon kojeg se obavlja isti proces pribrajanja vrijednosti rezidualne veze i normalizacije. Koderski slojevi se mogu nizati jedan za drugim kako bi se povećao kapacitet modela. Dekoderski sloj transformera je autoregresivan, što znači da pri predikciji sljedećeg člana izlaznog niza koristi informacije o ulaznom nizu u obliku latentnih vektorskih reprezentacija iz koderskog sloja, te informacije o svim prijašnjim članovima izlaznog niza. Generiranje članova izlaznog niza počinje tako da se na ulaz dekodera dovede poseban <START> token koji označava početak niza. Prvi dio dekoderskog sloja sastoji se od sloja višestruke pažnje koji je isti kao i u koderskom dijelu osim što koristi postupak maskiranja nakon skaliranja matrice umnoška upita i ključeva. Postupak maskiranja vrijednost svih članova matrice iznad glavne dijagonale postavlja na -∞ kako one ne bi imale utjecaja kod mehanizma pažnje. Razlog za obavljanje maskiranja je taj što riječi generirane u izlaznom nizu ne smiju moći obraćati pažnju na riječi koje su generirane nakon njih. Zatim slijedi drugi sloj višestruke pažnje koji se od prvog razlikuje po tome što za dobivanje ključeva i upita koristi latentne vektorske reprezentacije dobivene u koderskom sloju, a za dobivanje vrijednosti koristi latentne vektorske reprezentacije prijašnjeg sloja dekodera. To omogućuje da se pri generiranju izlaznog niza koristi mehanizam pažnje između riječi ulaznog i izlaznog niza. Izlaz drugog sloja višestruke pažnje se nakon dodavanja vrijednosti rezidualne veze i normalizacije propušta kroz potpuno povezani sloj koji je isti kao i u koderskom sloju. Normalizirani izlaz se zatim šalje u konačni potpuno povezani sloj čija je izlazna dimenzija jednaka dimenziji ukupnog korištenog vokabulara. Nad tim izlazom se primjenjuje aktivacijska funkcija softmax kako bi se dobila vjerojatnosna distribucija riječi u vokabularu, te se kao sljedeća riječ izlaznog niza odabire ona kojoj odgovara indeks člana s najvećom vjerojatnosti. Ta riječ se dodaje ulazu dekoderskog sloja pri generiranju sljedeće riječi izlaznog niza. Ovaj postupak se ponavlja sve dok se

ne generira poseban <END> token koji označava kraj niza. Ovi modeli pružaju veliki napredak u odnosu na povratne neuronske mreže i u pogledu efikasnog treniranja jer omogućuju propuštanje čitavog ulaznog niza kroz model u jednom prolazu. Kod povratnih neuronskih mreža je potrebno učiniti onoliko prolaza kroz model koliko ulazni niz sadrži riječi ili tokena.

1.2 Veliki jezični modeli

Arhitektura transformera omogućila je najveći napredak u obradi prirodnog jezika i na njoj se temelje današnji najbolji veliki jezični modeli. Ti modeli sastoje se od velikog broja transformerskih slojeva u nizu, sadrže stotine milijardi parametara i trenirani su korištenjem skupova podataka koji sadrže bilijune tekstnih tokena. Model Llama 3 tvrtke Meta treniran je na skupu podataka od 15 bilijuna tokena [4] što je prema procjenama samo red veličina manje od količine tokena u svom kvalitetnom javno dostupnom tekstu [5]. Treniranjem i testiranjem takvih velikih modela uočena su brojna izranjajuća svojstva (*eng. emergent ability*). Izranjajuća svojstva su svojstva koja su prisutna kod modela s velikim brojem parametara, a nisu prisutna kod modela s malim brojem parametara, odnosno svojstva koja se ne mogu predvidjeti ekstrapoliranjem iz poznatih zakona skaliranja [6]. Jedno od najznačajnijih izranjajućih svojstava je mogućnost rješavanja zadataka na temelju nekoliko primjera riješenih zadataka zadanih u korisničkom upitu prije samog zadatka (*eng. few-shot prompting*) bez dodatnog podešavanja (*eng. fine-tuning*) modela nakon predtreniranja. Primjer takvog upita koji demonstrira mogućnosti prilagodbe modela korištenja ovog svojstva za rješavanje raznih zadataka prikazan je na slici (Slika 1.3).



Slika 1.3 Primjer upita prilagođenog za određivanje sentimenta komentara

Ovo svojstvo prvi put je primjećeno kod modela GPT-3 koji sadrži 175 milijardi parametara i koji je zatim korišten kao temelj servisa ChatGPT [7].

2 Sigurnost velikih jezičnih modela

Sve navedene mogućnosti velikih jezičnih modela sa sobom donose brojne ranjivosti koje ih čine podložnim raznim napadima. Dodatnu razinu rizika unosi i činjenica da su ulazni podaci velikih jezičnih modela korisnički upiti napisani prirodnim jezikom, a prostor stanja takvih upita je gotovo beskonačan.

2.1 Rizici

Glavni rizici korištenja velikih jezičnih modela mogu se podijeliti na 5 glavnih kategorija i njihovih 12 potkategorija [8]. Glavne kategorije rizika su:

- rizici zloćudnih korisnika
- rizici u interakciji čovjeka i velikog jezičnog modela
- informacijski rizici
- rizici diskriminacije, isključivanja, toksičnosti, mržnje i uvredljivosti
- dezinformacijski rizici.

Rizici zloćudnih korisnika odnose se na namjerno korištenje velikih jezičnih modela kako bi se postigao zloćudan cilj, a dodatno se mogu podijeliti na 3 potkategorije. Prva od tih potkategorija je poticanje dezinformacijskih kampanja koja obuhvaća korištenje velikih jezičnih modela u kreiranju lažnog i ometajućeg sadržaja. Druga potkategorija je pomaganje u ilegalnim aktivnostima i obuhvaća korištenje velikih jezičnih modela kao izvora znanja i prijedloga o postupcima i radnjama koje su protuzakonite. Posljednja potkategorija je poticanje na neetičke i nesigurne radnje u sklopu koje se veliki jezični modeli koriste za kreiranje sadržaja koji nanosi štetu tuđem ugledu ili promovira nesigurne zdravstvene savjete.

Rizici u interakciji čovjeka i velikog jezičnog modela mogu se podijeliti na dvije potkategorije; rizike mentalnog zdravlja i rizici odnošenja prema velikom jezičnom modelu kao prema ljudskoj osobi.

Informacijski rizici korištenja velikih jezičnih modela dijele se na rizike otkrivanja osobnih informacija i rizike otkrivanja osjetljivih organizacijskih informacija. Prva potkategorija obuhvaća slučajeve u kojima veliki jezični model otkriva osobne informacije koje su pogreškom bile prisutne u skupu podataka na kojem je model treniran ili do kojih dolazi u sklopu sustava kojim se poboljšavaju njegove

sposobnosti odgovaranja na specifična domenska pitanja, kao što je generacija potpomognuta dohvaćanjem (*eng. retrieval-augmented generation, RAG*). Druga potkategorija istovjetna je prvoj uz razliku da se radi o podacima koji su povjerljivi i pripadaju određenoj organizaciji.

Rizici diskriminacije, isključivanja, toksičnosti, mržnje i uvredljivosti sadrže 3 potkategorije; rizike socijalnih stereotipova i nepoštene diskriminacije, rizike govora mržnje i rizike generiranja neprimjerenog sadržaja. Sve potkategorije se odnose na slučajeve u kojima veliki jezični modeli generiraju nepoželjan sadržaj iz imena potkategorije.

Dezinformacijski rizici dijele se na rizike širenja lažnog ili obmanjujućeg sadržaja i rizike stvaranja materijalne štete zbog širenja dezinformacija. Prva potkategorija odnosi se na općenite slučajeve u kojima veliki jezični modeli generiraju lažne vijesti, šire neutemeljene glasine ili neispravno interpretiraju činjenice, dok druga potkategorija obuhvaća slučajeve u kojim veliki jezični model generira sadržaj koji ima direktne posljedice za korisnika, kao što su financijski, pravni i medicinski savjeti.

2.2 Strategije napada

Strategije napada na velike jezične modele oslanjaju se na ključna obilježja modela i mogu se podijeliti u 4 skupine [9]. Prva skupina sadrži strategije napada koje se temelje na sklonosti modela nadopunjavanju korisničkog ulaznog teksta. Ova sklonost proizlazi iz postupka treniranja modela koji prati njihovu autoregresivnost, odnosno mogućnost predviđanja sljedećeg tokena na temelju prijašnjeg konteksta koji sadrži ulazni niz i prethodno predviđene tokene. Ovoj skupini pripadaju napadi afirmativnim sufiksima u kojima se odbijanje odgovaranja na štetno pitanje pokušava izbjeći dodavanjem kratkog afirmativnog teksta koji podsjeća na početak odgovora u kojem je pitanje prihvaćeno na kraj korisničkog upita, napadi promjenom konteksta u kojima se štetan upit pokušava maskirati dodavanjem okolnog teksta koji nije štetan, i napadi učenjem iz konteksta u kojima se u korisnički upit dodaju primjeri davanja odgovora na štetna pitanja kako bi se iskoristilo izranjajuće svojstvo učenja na temelju primjera u upitima.

Sljedeća skupina napada oslanja se na sklonost modela praćenju pravila, koja proizlazi iz uobičajenih postupaka podešavanja modela. Poznati napadi iz ove

skupine su napadi eufemizmima u kojima se štetan upit pokušava indirektno ili zamršeno napisati kako bi se izbjegli mehanizmi detekcije i otežalo modelovo prepoznavanje štetnosti, napadi ograničavanjem odgovora u kojima se unutar upita postavljaju uvjeti na odgovor modela kojima je cilj potaknuti željeni odgovor, i napadi simulacijom u kojima se upit postavlja u kontekstu izmišljene situacije čime se modelu otežava prepoznavanje štetnosti upita.

Treća skupina napada pokušava iskoristiti razne formate tekstualnih podataka koji su korišteni za treniranje velikog jezičnog modela i njihovom uporabom zaobići mehanizme detekcije i odbijanje odgovaranja. Napadi iz ove skupine uključuju napade prevođenjem u kojima se upiti prevode na jezike koji su slabo zastupljeni u skupu podataka na kojima je model treniran i na kojem se provodilo sigurnosno podešavanje modela, napade šifriranjem u kojima se upiti šifriraju ili enkodiraju na razne načine, i napade u kojima se kroz igru igranja uloga pokušava dovesti model u stanje u kojem je vjerojatnije da će dati željeni odgovor na štetan upit. Napadi iz ove skupine često su vrlo složeni pa zahtijevaju visoku razinu sposobnosti velikog jezičnog modela kako bi bili uspješni.

Posljednja skupina napada oslanja se na izravnu manipulaciju velikih jezičnih modela u situacijama kada korisnik ima potpuni pristup modelima. Napadi iz ove skupine koriste mijenjanje hiperparametara i parametara velikih jezičnih modela kako bi se zaobišli podešeni mehanizmi zaštite i detekcije štetnih upita, a ponekad koriste i direktno podešavanje modela na štetnim skupovima podataka.

Mogućnost korištenja i stopa uspješnosti navedenih napada ovise o mnogim faktorima kao što su sposobnosti modela, skup podataka na kojem je model treniran, postupcima sigurnosnog podešavanja modela i mnogi drugi. Ti faktori su specifični za svaki pojedini model i iz tog razloga je poznavanje točnog velikog jezičnog modela na kojem se temelji sustav jedna od najvažnijih informacija za napadače.

3 Skup podataka

Za pokušaj rješavanja zadatka detekcije velikih jezičnih modela na temelju njihovih odgovora potrebno je prikupiti reprezentativan skup podataka. Skup podataka mora se sastojati od raznih vrsta pitanja koja pokrivaju mnoga tematska područja, te odgovora koje različiti veliki jezični modeli generiraju na upite koji sadržavaju ta pitanja.

3.1 Pitanja

Pitanja sadržana u prikupljenom skupu podataka mogu se podijeliti na sljedeće kategorije:

- pitanja općeg znanja otvorenog tipa
- pitanja općeg znanja zatvorenog tipa
- pitanja logičkog zaključivanja
- pitanja parafraziranja
- pitanja sažimanja teksta.

Kategorije pitanja izabrane su s ciljem pokrivanja širokog spektra primjena velikih jezičnih modela i provjere različitih mogućnosti modela. Kategorije su također ograničene na primjene za koje je razumno očekivati da će biti omogućene kod većine agenata ili sustava temeljenih na velikim jezičnim modelima. Sva pitanja prikupljena su iz javno dostupnih skupova podataka.

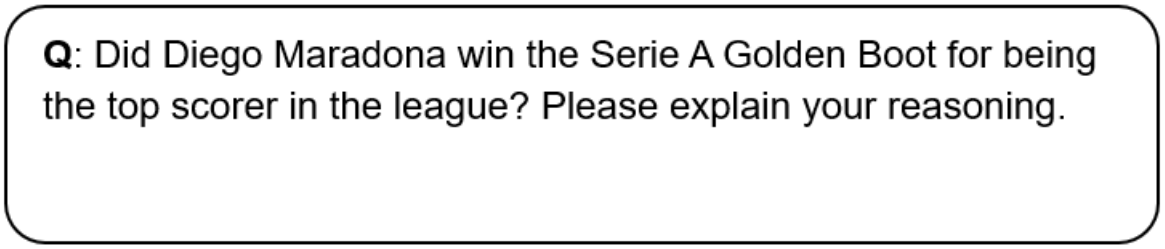
Pitanja općeg znanja otvorenog tipa prikupljena su iz skupa podataka General-Knowledge[1] koji se temelji na skupu podataka Stanford Alpaca [10]. Skup podataka sastoji se od 37600 pitanja i odgovora i originalno je namijenjen za treniranje velikih jezičnih modela. U skupu podataka zastupljene su brojne kategorije pitanja, a njihovi udjeli vidljivi su u tablici (Tablica 1).

[1] https://huggingface.co/datasets/MuskumPillerum/General-Knowledge

Tablica 1 Udio kategorija pitanja u skupu podataka General-Knowledge

Kategorija	Udio (%)
Priroda	16.5
UI, računarska znanost i robotika	7.3
Fizika i kemija	16.3
Geografija i povijest	11.2
Ljudi	16
Sport	13.5
Preporuke i dileme	17.8
Ostalo	1.4

Iz skupa podataka odabrano je 300 nasumičnih pitanja koristeći implementaciju Random[2] modula programskog jezika Python. Na kraj svakog pitanja konkateniran je tekst „Please explain your reasoning.“ kako bi se izbjegli kratki, neinformativni odgovori modela. Primjer pitanja iz ove kategorije vidljiv je na slici (Slika *3.1*).

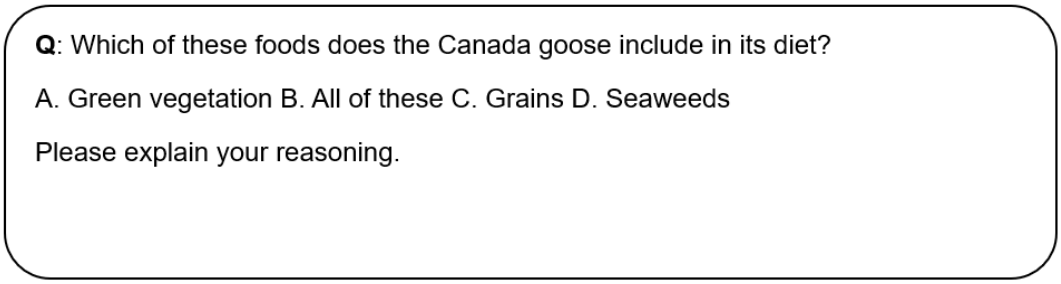


Slika 3.1 Primjer pitanja općeg znanja otvorenog tipa

Pitanja općeg znanja zatvorenog tipa prikupljena su iz skupa podataka OpenTriviaQA[3]. Skup pitanja sastoji se od 22 tematske kategorije, kao što su „animals“, „entertainment“, „movies“ i mnoge druge. Svako pitanje sadrži tekst, ponuđene odgovore i označeni točan odgovor. Kod prikupljanja pitanja zanemarena je kategorija „for-kids“ iz razloga što su pitanja unutar te kategorije vrlo jednostavna i ne bi značajno doprinijela provjeravanju znanja modela. Iz svake od preostalih kategorija nasumično je odabrano 15 pitanja koja su sadržavala 4 ponuđena odgovora. Na kraj svakog pitanja konkateniran je tekst

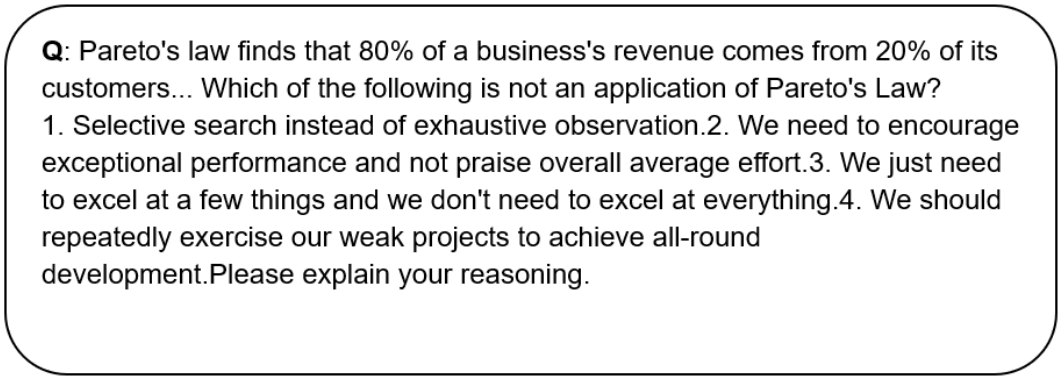
[2] https://python.readthedocs.io/en/stable/library/random.html
[3] https://github.com/uberspot/OpenTriviaQA

„Please explain your reasoning.“. Primjer pitanja iz ove kategorije vidljiv je na slici (Slika *3.2*).



Slika 3.2 Primjer pitanja općeg znanja zatvorenog tipa

Za pitanja logičkog zaključivanja korišten je skup podataka LogiQA [11]. Skup podataka sadrži pitanja korištena za provjeru čitanja s razumijevanjem. Svaki podatak sadrži informativni tekst, pitanje o tekstu i nekoliko ponuđenih odgovora. Iz skupa podataka je odabrano nasumičnih 300 pitanja i na kraj svakog pitanja je konkateniran tekst „Please explain your reasoning.“ Primjer pitanja iz ove kategorije vidljiv je na slici (Slika *3.3*).

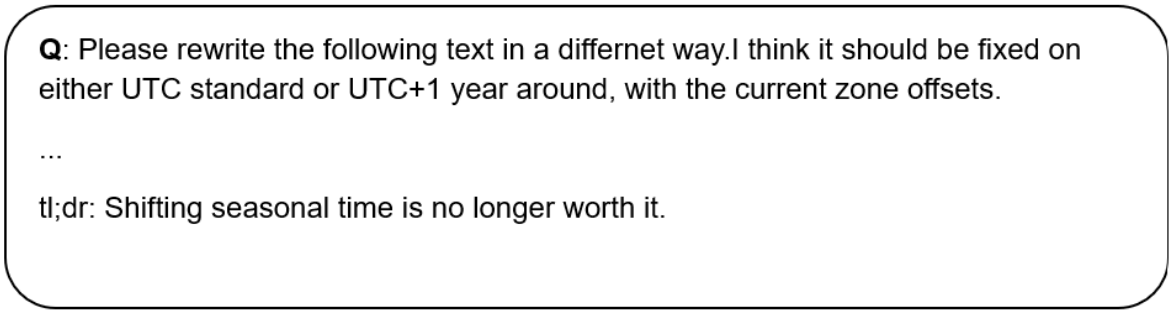


Slika 3.3 Primjer pitanja logičkog zaključivanja

Pitanja parafraziranja sastoje se od teksta kojem je na početak dodana uputa „Please rewrite the following text in a different way.” Za prikupljanje tekstova korišten je skup za treniranje skupa podataka Reddit [12][4]. Skup se sastoji od 3848330 objava na društvenoj mreži Reddit[5], a svaka sadrži ime autora, tekst objave i ime Subreddit-a. Iz skupa podataka je nasumično odabrano 300 objava, pri čemu je pažnja obraćena na to da svaka objava bude iz zasebnog Subreddit-a

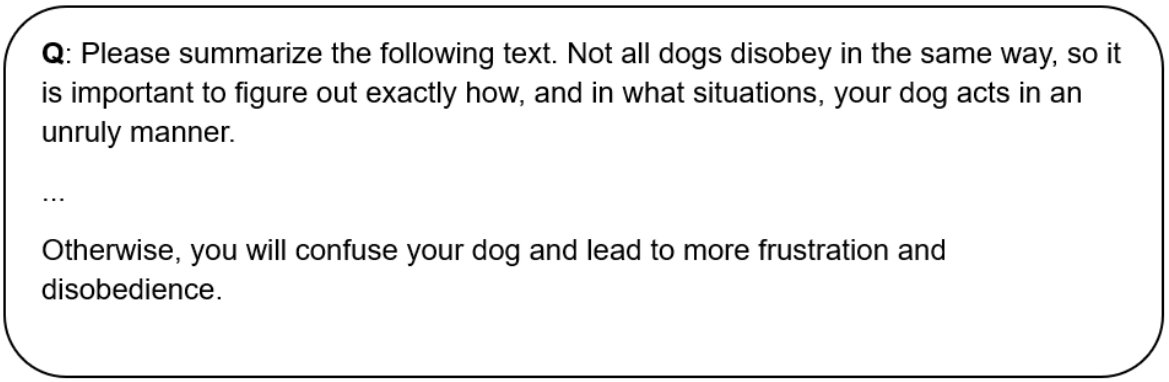
[4] https://huggingface.co/datasets/webis/tldr-17
[5] https://www.reddit.com/

kako bi se postigla što veća raznovrsnost tema. Primjer pitanja iz ove kategorije vidljiv je na slici (Slika *3.4*).



Slika 3.4 Primjer pitanja parafraziranja

Za pitanja sažimanja teksta korišten je WikiHow [13] skup podataka. Skup podataka sadrži članke objavljene na popularnoj WikiHow[6] stranici, u kojima se nalaze objašnjenja rješenja raznih problema. Iz skupa podataka odabrano je 300 nasumičnih tekstova i na početak im je dodan tekst „Please summarize the following text.“. Primjer pitanja iz ove kategorije vidljiv je na slici (Slika *3.5*).



Slika 3.5 Primjer pitanja sažimanja teksta

Konačan skup pitanja sastojao se od 1515 pitanja, po 300 iz svake kategorije osim pitanja općeg znanja zatvorenog tipa iz koje je uzeto 315 pitanja.

3.2 Odgovori

Slijedeći korak u prikupljanju skupa podataka bilo je dobivanje odgovora velikih jezičnih modela na pitanja iz prošlog koraka. Odabrani su neki od najpoznatijih velikih jezičnih modela kako bi skup podataka bio relevantan stvarnim primjenama.

[6] https://www.wikihow.com/Main-Page

Veliki jezični modeli odabrani u sklopu ovog rada su:

- gpt-3.5-turbo-1106
- gpt-3.5-turbo-0125
- mistral-medium-latest
- claude-3-sonnet-20240229
- llama-2-70b-chat.

Modeli gpt-3.5-turbo-1106 i gpt-3.5-turbo-0125 verzije su modela GPT-3.5[7] tvrtke OpenAI[8]. Duljina kontekstnog prozora (eng. context window) obje verzije modela je 16385 tokena, a zadnji datum podataka na kojem su modeli trenirani, poznat kao granica znanja (*eng. knowledge cutoff*), je rujan 2021. godine. Odabrane su obje verzije kako bi se kod klasifikacije mogla testirati mogućnost klasifikatora da razlikuje različite verzije istog modela. Odgovori obje verzije modela na pitanja dobiveni su korištenjem API-ja dostupnog u OpenAI Python biblioteci[9].

Model mistral-medium-latest posljednja je verzija srednje velikog modela tvrtke Mistral[10]. Duljina kontekstnog prozora ovog modela je 32000 tokena, a granica znanja modela je listopad 2022. godine. Odgovori ovog modela dobiveni su korištenjem API-ja dostupnog u Mistral Python biblioteci[11].

Model claude-3-sonnet-20240229 verzija je modela Claude 3[12] tvrtke Anthropic[13]. Duljina kontekstnog prozora ovog modela je 200000 tokena, a granica znanja modela je kolovoz 2023. godine. Odgovori ovog modela dobiveni su korištenjem API-ja dostupnog u Anthropic Python biblioteci[14].

Model llama-2-70b-chat verzija je modela Llama 2[15] tvrtke Meta. Za razliku od ostalih korištenih modela, parametri ovog modela su javno dostupni. Duljina kontekstnog prozora ovog modela je 4096 tokena, a granica znanja modela je prosinac 2023. godine. Zbog kratke duljine kontekstnog prozora ovog modela iz

[7] https://platform.openai.com/docs/models/gpt-3-5-turbo
[8] https://openai.com/
[9] https://pypi.org/project/openai/0.26.5/
[10] https://mistral.ai/
[11] https://github.com/mistralai/client-python
[12] https://www.anthropic.com/news/claude-3-family
[13] https://www.anthropic.com/
[14] https://pypi.org/project/anthropic/0.2.5/
[15] https://llama.meta.com/llama2/

skupa pitanja izbačena su 4 pitanja sažimanja teksta koja su bila predugačka. Budući da su parametri ovog modela javno dostupni, ne postoji javno dostupan API za dobivanje odgovora modela. Iz tog razloga je za dobivanje odgovora korišten servis deepinfra[16] na čijoj je infrastrukturi model pokrenut i pozivan pomoću omatanja API poziva API-jem OpenAI Python biblioteke.

Skup podataka se u konačnici sastoji od 7555 parova pitanja i odgovora, 1511 za svaki model zbog izbacivanja 4 pitanja koja su predugačka za llama-2-70b-chat model. Pri pozivanju API-ja za generiranje odgovora modela parametar *max_tokens* postavljen je na 300, a parametar *temperature,* kojim se regulira kreativnost i slučajnost generiranja odgovora, postavljen je na 0.2. Svaki par pitanja i odgovora sadrži oznaku modela niske razine (eng. low level model), koja sadrži točan naziv modela koji je generirao odgovor, i oznaku modela visoke razine (eng. high level model) u sklopu koje su modeli gpt-3.5-turbo-1106 i gpt-3.5-turbo-0125 spojeni u skupinu gpt-3.5.

[16] https://deepinfra.com/

4 Klasifikacija

U sljedećem koraku cilj je bio stvoriti sustav klasifikacije velikih jezičnih modela na temelju njihovih odgovora. Osnovna ideja temelji se na pretpostavci da bi svaki veliki jezični model trebao imati svojstveni stil generiranja odgovora. Ova pretpostavka oslanja se na samo treniranje modela, odnosno na činjenicu da se za svaki pojedini model pri treniranju koristio različit skup podataka, različiti postupak treniranja, pa čak i različiti redoslijed dohvaćanja primjera za treniranja. Svi navedeni faktori utjecali su na konačan skup parametara modela koji igra ključnu ulogu tijekom generiranja odgovora.

4.1 Pretprocesiranje podataka i značajke

Prije same klasifikacije potrebno je dodatno pripremiti podatke i iz njih izvući potrebne značajke. Prvi korak pretprocesiranja bio je lematizacija (*eng. lemmatization*) teksta odgovora. Lematizacija se često definira kao postupak uklanjanja nastavaka riječi dodanih tijekom mijenjanja gramatičkih kategorija, korištenjem rječnika i morfološke analize [14]. Osnovni oblik riječi koji dobivamo lematizacijom naziva se lema ili rječnički oblik (*eng. dictionary form*).

Slijedeći korak bio je dobivanje latentnih vektorskih reprezentacija (*eng. embedding*) teksta koje će biti korištene kao značajke kod klasifikacije. Ovdje su korištene dvije strategije dobivanja latentnih vektorskih reprezentacija. U prvoj strategiji se latentna vektorska reprezentacija odgovora računala kao srednja vrijednost latentnih vektorskih reprezentacija svih riječi u odgovoru, dobivenih korištenjem „en_core_web_lg“ modula Pythonove spaCy[17] biblioteke. Ove reprezentacije su 300-dimenzionalne i zbog uzimanja srednje vrijednosti sadrže nepotpunu kumulativnu informaciju o sadržaju odgovora. U drugoj strategiji se za dobivanje latentne vektorske reprezentacije odgovora koristio model Text-Embedding-3-Large[18] tvrtke OpenAI. Ove reprezentacije su 3072-dimenzionalne i sadržavaju potpunu kumulativnu informaciju o sadržaju teksta jer se za dobivanje koristi cijeli tekst odgovora odjednom.

[17] https://spacy.io/
[18] https://platform.openai.com/docs/models/embeddings

4.2 Klasifikatori

Za klasifikatore su odabrana 3 modela:

- BERT
- stroj potpornih vektora
- potpuno povezana unaprijedna neuronska mreža.

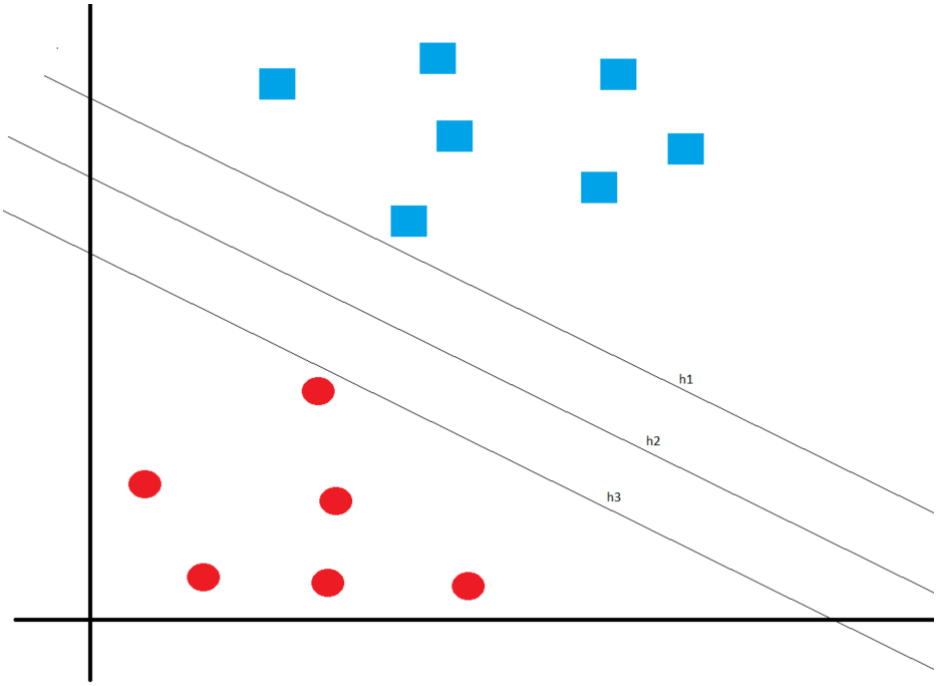
BERT (Bidirectional Encoder Representations from Transformers) model [15] je višeslojni dvosmjerni transformer. Riječ „dvosmjeran“ odnosi se na napredak modela u odnosu na originalnu arhitekturu transformera koji mu omogućuje da koristi i lijevi i desni kontekst riječi. Model se sastoji od 12 transformerskih slojeva i 12 slojeva pažnje (*eng. attention head*). Broj dimenzija skrivenog sloja modela je 768, a model ukupno sadrži 110 milijuna parametara. Model je predtreniran na dva nenadzirana zadatka. Prvi zadatak je maskirano jezično modeliranje (*eng. masked language modeling*) u kojem je zadatak predvidjeti token koji se nalazi na mjestu posebnog [MASK] tokena na ulazu modela. Za vrijeme ovog zadatka se 15% nasumično odabranih tokena maskira na način da ih se 80% zamjeni [MASK] tokenom, 10% nasumičnim tokenom, a ostalih 10% ostane nepromijenjeno. Razlog za provođenje ove procedure u kojoj se [MASK] tokenom ne zamjenjuju svi nasumično odabrani tokeni je taj što se [MASK] token ne pojavljuje za vrijeme prilagođavanja (*eng. fine-tuning*) modela, pa dolazi do nepodudarnosti zadataka. Zadatak maskiranog jezičnog modeliranja omogućuje dvosmjerno djelovanje mehanizma pažnje, odnosno korištenje lijevog i desnog konteksta maskiranog tokena kako bi se predvidjela njegova vrijednost. U konačnici se skrivena vektorska reprezentacija maskiranog tokena koristi za predviđanje njegove originalne vrijednosti. Drugi zadatak je predviđanje sljedeće rečenice (*eng. next sentence prediction*). U sklopu tog zadatka se na ulaz modela dovode rečenice *A* i *B* odvojene posebnim [SEP] tokenom, a zatim se pomoću posljednje skrivene vektorske reprezentacije posebnog [CLS] tokena koji se postavlja na početak svakog ulaznog niza predviđa slijedi li rečenica *B* nakon rečenice *A* u originalnom tekstu. Tijekom predtreniranja je u 50% slučajeva rečenica *B* uistinu bila ona koja u originalu slijedi nakon rečenice *A*, dok je u drugih 50% slučajeva zamijenjena nasumičnom rečenicom. Za zadatak klasifikacije odgovora velikih jezičnih modela

pomoću BERT modela korištena je implementacija modela i pripadnog tokenizatora teksta iz HuggingFace[19] biblioteke.

Stroj potpornih vektora (*eng. Support Vector Machine*), ili SVM, slijedeći je klasifikator odabran za ovaj zadatak. Model SVM-a običan je linearni model zadan formulom (5):

$$h(x; w) = w^T x. \tag{5}$$

U formuli *w* predstavlja težine modela, a *x* značajke ulaznog primjera kojeg klasificiramo. Osnovna ideja modela SVM može se formalizirati kao problem maksimalne margine. Margina je udaljenost od (n-1)-dimenzionalne hiperravnine, koja odvaja primjere klasa u n-dimenzionalnom prostoru, do najbližeg primjera sa svake strane. Maksimizacijom margine sprječava se pristranost kod klasifikacije i postiže se bolja generalizacija modela. Na slici (Slika 4.1) prikazan je koncept maksimalne margine. Iako sve 3 hipoteze, od beskonačno mnogo mogućih, savršeno odvajaju primjere dviju klasa, hipoteza h_2 najmanje je pristrana zato što se nalazi na maksimalnoj udaljenosti od primjera.



Slika 4.1 Maksimalna margina

[19] https://huggingface.co/google-bert/bert-base-uncased

Zbog optimizacije je korisno iz primarne formulacije, u kojoj su primarni parametri težine, model prebaciti u dualnu formulaciju (6):

$$h(x) = \sum_{i=1}^{N} \alpha_i y^i x^T x^i \; + \; w_{0,} \tag{6}$$

u kojoj su parametri faktori *α*. U dualnoj formulaciji se za svaki primjer *x* koji se želi klasificirati računa skalarni umnožak pomnožen koeficijentom α_i i oznakom y^i sa svakim primjerom x^i iz skupa za treniranje. Optimizacijski uvjet komplementarne labavosti (7):

$$\alpha_i(y^i h(x^i) - 1) = 0 \tag{7}$$

osigurava da su koeficijenti α_i različiti od 0 samo za primjere kod kojih je $y^i h(x^i)$*=1*, a to su upravo primjeri koji se nalaze na samoj margini, odnosno primjeri koji su najbliži optimalnoj hipotezi. Ti primjeri, koji jedini utječu na klasifikaciju novog primjera, nazivaju se potporni vektori. SVM model se dodatno poopćava na skupove primjera koji nisu linearno odvojivi uvođenjem meke margine. Meka margina relaksira uvijete optimizacije tako da dozvoljava, ali i kažnjava, mogućnosti da se primjeri nalaze unutar margine ili na strani hipoteze na kojoj su pogrešno klasificirani, u svrhu bolje generalizacije. Skalarni umnožak u modelu i optimizacijskom postupku zapravo predstavlja mjeru sličnosti između primjera. Korištenjem skalarnog umnoška dobivamo decizijsku granicu koja je linearna u prostoru značajki. Ovo ograničenje zaobilazimo korištenjem takozvanog „jezgrenog trika“ [16], odnosno korištenjem raznih jezgrenih funkcija umjesto skalarnog umnoška u modelu SVM-a i optimizacijskom postupku. Jezgrene funkcije *K* su funkcije koje računaju sličnost između dva primjera i za njih vrijedi:

$$K(x_1, x_2) \geq 0 \tag{8}$$

i

$$K(x_1, x_2) = K(x_2, x_1). \tag{9}$$

SVM je, kao i mnogi drugi modeli, pogodan isključivo za binarnu klasifikaciju. Kod takvih modela se višeklasna klasifikacija ostvaruje korištenjem posebnih strategija. Prva od tih strategija naziva se „jedna protiv ostalih“ (*eng. one versus rest, OVR*). U toj strategiji se, kao što joj i ime govori, za svaku od ciljnih klasa trenira zaseban klasifikator na način da se primjeri te klase i primjeri svih ostalih klasa odvoje u dvije klase kako bi se primijenila binarna klasifikacija. Druga strategija naziva se

„jedna protiv jedne“ (*eng. one versus one, OVO*). Za razliku od prve strategije, ovdje se za svaki par ciljnih klasa trenira zaseban binarni klasifikator. Kod obje strategije se konačna predikcija dobiva metodom glasovanja (*eng. voting*) koristeći predikcije svih uključenih klasifikatora. U sklopu rada je korištena implementacija SVM klasifikatora iz Pythonove Scikit-learn biblioteke[20].

Potpuno povezana unaprijedna neuronska mreža sastojala se od ulaznog sloja, srednjeg sloja koji je sadržavao 100 neurona i izlaznog sloja čija je dimenzionalnost bila jednaka broju klasa u zadatku klasifikacije. Aktivacijska funkcija korištena u srednjem sloju je zglobnica (*eng. rectified linear unit, ReLU*). Potpuno povezana unaprijedna neuronska mreža implementirana je koristeći programski okvir PyTorch[21].

4.3 Treniranje modela

Svi modeli trenirani su na tri različita zadatka. Prvi zadatak bio je klasifikacija velikog jezičnog modela niske razine, drugi klasifikacija velikog jezičnog modela visoke razine, a treći binarna klasifikacija svakog pojedinog modela. Kod trećeg zadatka se svaki pojedini model klasificira strategijom „jedna protiv ostalih“. Skup podataka za treniranje sadržavao je 80% ukupnog broja podataka, dok je preostalih 20% služilo kao skup za testiranje. Kod podjele skupa podataka bilo je nužno obratiti pozornost na nekoliko uvjeta. Prvi je bio taj da je nužno u skupu podatak za treniranje i skupu podataka za testiranje imati jednak broj primjera koji pripadaju svakoj klasi, odnosno svakom velikom jezičnom modelu. Drugi je bio taj da je potrebno u skupu za treniranje i skupu za testiranje imati podjednak omjer pitanja iz svih 5 kategorija. Sama struktura skupa podataka pomogla je u ostvarenju navedenih uvjeta jer su odgovori svih modela prikupljeni zasebno i zatim konkatenirani u konačan skup podataka, a pitanja su bila poredana istim redoslijednom po kategorijama kod svih modela. Zbog takve strukture bilo je dovoljno nasumično izmiješati indekse pitanja, odvojiti prvih 1208 indeksa pitanja koji čine 80% ukupnog broja i pripadaju skupu za treniranje, odvojiti preostalih 303 indeksa pitanja koji čine 20% ukupnog broja i pripadaju skupu za testiranje, i zatim

[20] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
[21] https://pytorch.org/

dodati pitanja koja se nalaze na tim indeksima u pripadajuće skupove za svaki model.

BERT model podešavan je 10 epoha na zadacima klasifikacije modela visoke i niske razine, a na zadacima binarne klasifikacije podešavan je 5 epoha. Korišten je optimizator Adam sa stopom učenja 10^{-5} i veličinom grupe (*eng. batch size*) od 16 primjera. Unakrsna entropija je korištena kao funkcija gubitka. Za klasifikaciju pomoću BERT modela koristi se posljednja skrivena vektorska reprezentacija posebnog [CLS] tokena ulaznog niza. Ta reprezentacija se zatim proslijedi kao ulaz u potpuno povezani sloj čija je izlazna dimenzija jednaka broju klasa u zadatku. Kao ulaz modela korišten su tokeni dobiveni pripadajućim BERT tokenizatorom iz originalnih odgovora velikih jezičnih modela.

Kod treniranja SVM modela korištene su različite značajke. Prvo su korišteni 300-dimenzionalni vektori srednjih vrijednosti latentnih reprezentacija svih riječi u odgovorima. Zatim su korištene latentne vektorske reprezentacije dobivene koristeći model Text-Embedding-3-Large tvrtke OpenAI od 3072 dimenzija. Nakon toga su korištene reprezentacije originalnog i lematiziranog teksta dobivene vektorizacijom pomoću CountVectorizer-a[22] iz biblioteke Scikit-learn. Reprezentacija teksta dobivena na ovaj način naziva se model vreće riječi (*eng. bag of words model*) jer kao značajke sadrži broj pojavljivanja pojedinih tokena riječi u tekstu. Tokeni riječi se kod ovog načina modeliranja teksta nazivaju n-grami, gdje n označava broj riječi sadržanih u jednom tokenu. Posljednja vrsta značajki su reprezentacije originalnog i lematiziranog teksta dobivene vektorizacijom pomoću TfidfVectorizer-a[23] iz biblioteke Scikit-learn. TfidfVectorizer modelira tekst na sličan način kao i kod modela vreće riječi, ali umjesto broja pojavljivanja n-grama koristi njihovu TF-IDF (*eng. term frequency-inverse document frequency*) vrijednost. TF-IDF vrijednost n-grama svojevrsni je pokazatelj važnosti n-grama u tekstu. TF-IDF vrijednost računa se formulom (10):

$$TFIDF(t, d, D) = TF(t, d) \cdot IDF(t, D), \tag{10}$$

[22] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
[23] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

gdje *t* označava n-gram, *d* označava tekst u kojem se n-gram nalazi, a *D* označava skup svih tekstova koji se koriste pri vektorizaciji. Izraz *TF(t,d)* predstavlja učestalost pojavljivanja n-grama *t* u tekstu *d*, a to pokazuje koliko je pojam relevantan i bitan za sami tekst. Izraz *IDF(t,D)* predstavlja inverz učestalosti pojavljivanja n-grama u cijelom skupu tekstova *D*. Time se postiže da se pojmove koji su učestali u svim tekstovima, poput raznih veznika i čestih općenitih riječi, kod *TF-IDF* vektorizacije vrednuje manje nego riječi koje su učestale isključivo u tekstu *d* za koji se vektorizacija računa. Za optimizaciju hiperparametara SVM modela korišten je postupak mrežnog pretraživanja (*eng. grid search*). U sklopu mrežnog pretraživanja korišten je postupak peterostruke unakrsne validacije (*eng. 5-fold cross-validation*) kod kojeg se skup podataka 5 puta dijeli na skup za treniranje i skup za testiranje za svaku kombinaciju hiperparametara, te se zatim kao konačan rezultat uzima srednja vrijednost mjere vrednovanja modela kroz svih 5 postupaka treniranja. Hiperparametri modela SVM koji su optimizirani su regularizacijski faktor C, čija je vrijednost obrnuto proporcionalna jačini regularizacije i vrsta jezgrene funkcije. U situacijama u kojima je za dobivanje značajki korišten vektorizator dodatno je optimiziran i raspon n-grama koje vektorizator koristi. U tablici (Tablica 2) vidljive su sve isprobane vrijednosti hiperparametara.

Tablica 2 Isprobane vrijednosti hiperparametara

Hiperparametar	**Vrijednosti**
C	0.1, 1, 10
Jezgrena funckija	'linear', 'rbf'
Raspon n-grama	(1,1), (1,2), (1,3)

Potpuno povezana unaprijedna neuronska mreža trenirana je koristeći latentne vektorske reprezentacije odgovora dobivene koristeći model Text-Embedding-3-Large tvrtke OpenAI od 3072 dimenzija. Dimenzija izlaznog sloja modela bila je jednaka broju klasa u zadatku klasifikacije. Model je treniran 10 epoha koristeći početnu stopu učenja 10^{-3} koja je nakon 5 epoha smanjena na 10^{-4}. Za treniranje je korišten optimizator Adam, a kao funkcija gubitka korištena je unakrsna entropija.

4.4 Rezultati klasifikacije

Kao mjera vrednovanja modela odabrana je mjera uravnotežena točnost (*eng. balanced accuracy*)[24] kako je implementirana u biblioteci Scikit-learn, a rezultati su vidljivi u tablici (Tablica 3).

Tablica 3 Rezultati klasifikacije

	BERT	**Potpuno povezani model**	**SVM (spaCy reprezentacije)**	**SVM (Text-Embedding-3-Large reprezentacije)**	**SVM [25]**	**SVM [26]**	**SVM [27]**	**SVM [28]**
Klasifikacija niske razine	**0.675**	0.643	0.473	0.644	0.547	0.585	0.557	0.571
Klasifikacija visoke razine	**0.783**	**0.783**	0.562	**0.783**	0.664	0.671	0.654	0.678
Binarna klasifikacija GPT-3.5	**0.901**	0.823	0.777	0.842	0.758	0.830	0.756	0.835
Binarna klasifikacija GPT-3.5-1106	0.717	0.608	0.702	**0.751**	0.693	0.729	0.681	0.729
Binarna klasifikacija GPT-3.5-0125	0.701	0.618	0.702	**0.765**	0.719	0.750	0.700	0.741
Binarna klasifikacija Claude	**0.928**	0.831	0.841	0.874	0.911	0.886	0.915	0.887
Binarna klasifikacija Mistral	**0.776**	0.667	0.636	0.707	0.711	0.674	0.705	0.674
Binarna klasifikacija LLama	0.833	**0.991**	0.698	0.985	0.773	0.735	0.767	0.725

[24] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.balanced_accuracy_score.html
[25] Korištenje Tfidfvectorizer vektorizatora i originalnog teksta odgovora.
[26] Korištenje CountVectorizer vektorizatora I originalnog teksta odgovora.
[27] Korištenje Tfidfvectorizer vektorizatora i lematiziranog teksta odgovora.
[28] Korištenje CountVectorizer vektorizatora i lematiziranog teksta odgovora.

Mjera uravnotežene točnosti namijenjena je radu s nebalansiranim skupovima podataka, a računa se kao srednja vrijednost odziva (*eng. recall*) svake pojedine klase u zadatku klasifikacije.

U rezultatima možemo vidjeti da je zadatak klasifikacije modela niske razine teži od zadatka klasifikacije modela visoke razine. Glavni razlog je taj što dvije verzije modela GPT 3.5 daju vrlo slične odgovore. Također možemo primijetiti da je model Mistral značajno teže ispravno klasificirati od svih ostalih modela. U zadatku binarne klasifikacije za model Mistral je u najboljem slučaju postignuta uravnotežena točnost od 0.776 koristeći BERT model, dok su najbolji binarni klasifikatori za sve ostale modele, pri čemu obje verzije modela GPT 3.5 smatramo jednim modelom kao u trećem retku tablice (Tablica 3), postigli uravnoteženu točnost od barem 0.901.

5 Maskiranje modela

U sljedećem koraku ovog rada testirana je mogućnost maskiranja odgovora velikih jezičnih modela kako bi se onemogućila njihova detekcija. U sklopu ovog zadatka korišteni su najbolji klasifikatori iz prethodnog poglavlja za svaki pojedini zadatak klasifikacije, osim za binarnu detekciju modela Claude gdje je zbog manje resursne zahtjevnosti korišten SVM model usporedivih performansi s najboljim BERT modelom.

5.1 Metodologija

Za potrebe ovog zadatka svi najbolji klasifikatori trenirani su na istom skupu podatak za treniranje, dobivenim nasumičnim miješanjem liste indeksa s parametrom *random_seed* postavljenim na 9876. Performanse modela su zatim evaluirane na skupu za testiranje s originalnim i maskiranim odgovorima. Maskiranje odgovora implementirano je korištenjem jezičnog modela Phi-3 [17] tvrtke Microsoft. Ovaj model se naziva malim jezičnim modelom (*eng. small language model*) jer sadrži 3.8 milijardi parametara, što ga čini pogodnim za lokalno korištenje. Korištena je implementacija modela dostupna u biblioteci Ollama[29]. Odgovori velikih jezičnih modela su maskirani modelom Phi-3 tako da su konkatenirani na tekst „Rewrite the following text without altering the meaning, but change the syntax as much as possible: “ i predani na ulaz modela. Ideja maskiranja odgovora velikih jezičnih modela parafraziranjem koristeći mali jezični model temelji se na sljedećim pretpostavkama:

- kod virtualnih agenata i sustava u produkciji nužno je koristiti velike jezične modele zbog njihovog znanja i mogućnosti strukturiranja odgovora
- mali jezični modeli ne posjeduju jednaku količinu znanja, ali imaju dovoljno razvijene sposobnosti strukturiranja i razumijevanja teksta da mogu kvalitetno parafrazirati odgovore drugih modela.

5.2 Rezultati

U nastavku su prikazani i protumačeni rezultati testiranja najboljih klasifikatora na maskiranim i originalnim odgovorima. Za evaluaciju ovih modela korištene su

[29] https://ollama.com/library/phi3

mjere uravnotežene točnosti, preciznosti (*eng. precision*), odziva (*eng. recall*) i mjera F1.

Prvo je isproban najbolji model iz zadatka binarne klasifikacije modela Claude. Radi se o SVM modelu koji koristi nelematizirane odgovore, TF-IDF vektorizaciju s rasponom n-grama od 1 do 3, regularizacijski faktor C od 10 i linearnu jezgru. U tablicama (Tablica 4, Tablica 5) vidljivi su rezultati testiranja ovog modela.

Tablica 4 Rezultati testiranja originalnih odgovora modela Claude

	Preciznost	**Odziv**	**F1**	**Potpora**
Ostali modeli	0.97	0.94	0.96	1212
Claude	0.80	0.88	0.84	303
Mikro prosjek/točnost			0.93	1515
Makro prosjek	0.88	0.91	0.90	1515
Težinski prosjek	0.94	0.93	0.93	1515
Uravnotežena točnost	0.913			

Tablica 5 Rezultati testiranja maskiranih odgovora modela Claude

	Preciznost	**Odziv**	**F1**	**Potpora**
Ostali modeli	0.91	0.60	0.72	1212
Claude	0.32	0.75	0.45	303
Mikro prosjek/točnost			0.63	1515
Makro prosjek	0.61	0.67	0.58	1515
Težinski prosjek	0.79	0.63	0.67	1515
Uravnotežena točnost	0.674			

Sljedeći model je najbolji model iz zadatka binarne klasifikacije modela Llama. Radi se o potpuno povezanoj unaprijednoj neuronskoj mreži koja kao ulazne značajke koristi latentne vektorske reprezentacije teksta. U tablicama (Tablica 6, Tablica 7) vidljivi su rezultati testiranja ovog modela.

Tablica 6 Rezultati testiranja originalnih odgovora modela Llama

	Preciznost	Odziv	F1	Potpora
Ostali modeli	1.00	1.00	1.00	1212
Llama	1.00	0.98	0.99	303
Mikro prosjek/točnost			1.00	1515
Makro prosjek	1.00	0.99	0.99	1515
Težinski prosjek	1.00	1.00	1.00	1515
Uravnotežena točnost	0.991			

Tablica 7 Rezultati testiranja maskiranih odgovora modela Llama

	Preciznost	Odziv	F1	Potpora
Ostali modeli	0.80	1.00	0.89	1212
Llama	0.29	0.01	0.01	303
Mikro prosjek/točnost			0.80	1515
Makro prosjek	0.54	0.50	0.45	1515
Težinski prosjek	0.70	0.80	0.71	1515
Uravnotežena točnost	0.501			

Sljedeći model je najbolji model iz zadatka binarne klasifikacije modela GPT 3.5. Radi se o BERT modelu, a u tablicama (Tablica 8, Tablica 9) vidljivi su rezultati testiranja ovog modela.

Tablica 8 Rezultati testiranja originalnih odgovora modela GPT 3.5

	Preciznost	Odziv	F1	Potpora
Ostali modeli	0.93	0.91	0.92	909
GPT 3.5	0.86	0.89	0.88	606
Mikro prosjek/točnost			0.90	1515
Makro prosjek	0.90	0.90	0.90	1515
Težinski prosjek	0.90	0.90	0.90	1515
Uravnotežena točnost	0.899			

Tablica 9 Rezultati testiranja maskiranih odgovora modela GPT 3.5

	Preciznost	Odziv	F1	Potpora
Ostali modeli	0.77	0.79	0.78	909
GPT 3.5	0.68	0.64	0.66	606
Mikro prosjek/točnost			0.73	1515
Makro prosjek	0.72	0.72	0.72	1515
Težinski prosjek	0.73	0.73	0.73	1515
Uravnotežena točnost	0.712			

Sljedeći model je najbolji model iz zadatka binarne klasifikacije modela gpt-3.5-turbo-1106. Radi se o SVM modelu koji koristi latentne vektorske reprezentacije dobivene modelom Text-Embedding-3-Large, regularizacijski faktor C od 1 i RBF jezgru. U tablicama (Tablica 10, Tablica 11) vidljivi su rezultati testiranja ovog modela.

Tablica 10 Rezultati testiranja originalnih odgovora modela gpt-3.5-turbo-1106

	Preciznost	Odziv	F1	Potpora
Ostali modeli	0.95	0.66	0.78	1212
gpt-3.5-turbo-1106	0.39	0.86	0.53	303
Mikro prosjek/točnost			0.70	1515
Makro prosjek	0.67	0.76	0.66	1515
Težinski prosjek	0.84	0.70	0.73	1515
Uravnotežena točnost	0.759			

Tablica 11 Rezultati testiranja maskiranih odgovora modela gpt-3.5-turbo-1106

	Preciznost	Odziv	F1	Potpora
Ostali modeli	0.88	0.68	0.77	1212
gpt-3.5-turbo-1106	0.33	0.63	0.44	303
Mikro prosjek/točnost			0.67	1515
Makro prosjek	0.61	0.66	0.60	1515
Težinski prosjek	0.77	0.67	0.70	1515
Uravnotežena točnost	0.658			

Sljedeći model je najbolji model iz zadatka binarne klasifikacije modela gpt-3.5-turbo-0125. Radi se o SVM modelu koji koristi latentne vektorske reprezentacije

dobivene modelom Text-Embedding-3-Large, regularizacijski faktor C od 1 i RBF jezgru. U tablicama (Tablica 12, Tablica 13) vidljivi su rezultati testiranja ovog modela.

Tablica 12 Rezultati testiranja originalnih odgovora modela gpt-3.5-turbo-0125

	Preciznost	**Odziv**	**F1**	**Potpora**
Ostali modeli	0.94	0.68	0.79	1212
gpt-3.5-turbo-0125	0.39	0.84	0.54	303
Mikro prosjek/točnost			0.71	1515
Makro prosjek	0.67	0.76	0.66	1515
Težinski prosjek	0.83	0.71	0.74	1515
Uravnotežena točnost	0.758			

Tablica 13 Rezultati testiranja maskiranih odgovora modela gpt-3.5-turbo-0125

	Preciznost	**Odziv**	**F1**	**Potpora**
Ostali modeli	0.91	0.61	0.73	1212
gpt-3.5-turbo-0125	0.32	0.75	0.45	303
Mikro prosjek/točnost			0.63	1515
Makro prosjek	0.61	0.68	0.59	1515
Težinski prosjek	0.79	0.63	0.67	1515
Uravnotežena točnost	0.678			

Sljedeći model je najbolji model iz zadatka binarne klasifikacije modela Mistral. Radi se o BERT modelu, a u tablicama (Tablica 14, Tablica 15) vidljivi su rezultati testiranja ovog modela.

Tablica 14 Rezultati testiranja originalnih odgovora modela Mistral

	Preciznost	**Odziv**	**F1**	**Potpora**
Ostali modeli	0.89	0.89	0.89	1212
Mistral	0.57	0.57	0.57	303
Mikro prosjek/točnost			0.83	1515
Makro prosjek	0.73	0.73	0.73	1515
Težinski prosjek	0.83	0.83	0.83	1515
Uravnotežena točnost	0.733			

Tablica 15 Rezultati testiranja maskiranih odgovora modela Mistral

	Preciznost	Odziv	F1	Potpora
Ostali modeli	0.82	0.80	0.81	1212
Mistral	0.28	0.32	0.30	303
Mikro prosjek/točnost			0.70	1515
Makro prosjek	0.55	0.56	0.55	1515
Težinski prosjek	0.72	0.70	0.71	1515
Uravnotežena točnost	0.558			

Sljedeći model je najbolji model iz zadatka klasifikacije modela niske razine. Radi se o BERT modelu, a u tablicama (Tablica 16, Tablica 17) vidljivi su rezultati.

Tablica 16 Rezultati testiranja originalnih odgovora kod klasifikacije modela niske razine

	Preciznost	Odziv	F1	Potpora
claude-3-sonnet-20240229	0.93	0.90	0.91	303
gpt-3.5-turbo-0125	0.60	0.28	0.38	303
gpt-3.5-turbo-1106	0.50	0.72	0.59	303
llama-2-70b-chat	0.65	0.81	0.72	303
mistral-medium-latest	0.61	0.53	0.57	303
Mikro prosjek/točnost			0.65	1515
Makro prosjek	0.66	0.65	0.64	1515
Težinski prosjek	0.66	0.65	0.64	1515
Uravnotežena točnost	0.649			

Tablica 17 Rezultati testiranja maskiranih odgovora kod klasifikacije modela niske razine

	Preciznost	Odziv	F1	Potpora
claude-3-sonnet-20240229	0.38	0.77	0.51	303
gpt-3.5-turbo-0125	0.38	0.48	0.42	303
gpt-3.5-turbo-1106	0.55	0.06	0.11	303
llama-2-70b-chat	0.32	0.04	0.07	303
mistral-medium-latest	0.27	0.40	0.32	303
Mikro prosjek/točnost			0.35	1515
Makro prosjek	0.38	0.35	0.29	1515
Težinski prosjek	0.38	0.35	0.29	1515
Uravnotežena točnost	0.349			

Sljedeći model je najbolji model iz zadatka klasifikacije modela visoke razine. Radi se o potpuno povezanoj unaprijednoj neuronskoj mreži koja kao ulazne značajke koristi latentne vektorske reprezentacije teksta. U tablicama (Tablica 18, Tablica 19) vidljivi su rezultati testiranja ovog modela.

Tablica 18 Rezultati testiranja originalnih odgovora kod klasifikacije modela visoke razine

	Preciznost	**Odziv**	**F1**	**Potpora**
claude-3-sonnet-20240229	0.77	0.84	0.80	303
gpt-3.5	0.81	0.84	0.83	606
llama-2-70b-chat	0.99	0.97	0.98	303
mistral-medium-latest	0.56	0.49	0.52	303
Mikro prosjek/točnost			0.79	1515
Makro prosjek	0.78	0.78	0.78	1515
Težinski prosjek	0.79	0.79	0.79	1515
Uravnotežena točnost	0.783			

Tablica 19 Rezultati testiranja maskiranih odgovora kod klasifikacije modela visoke razine

	Preciznost	**Odziv**	**F1**	**Potpora**
claude-3-sonnet-20240229	0.33	0.70	0.45	303
gpt-3.5	0.73	0.53	0.61	606
llama-2-70b-chat	0.20	0.01	0.01	303
mistral-medium-latest	0.26	0.36	0.30	303
Mikro prosjek/točnost			0.43	1515
Makro prosjek	0.38	0.40	0.34	1515
Težinski prosjek	0.45	0.43	0.40	1515
Uravnotežena točnost	0.399			

U rezultatima je jasno vidljivo da su se maskiranjem smanjile performanse svih modela iz prethodnog zadatka. Razina smanjenja performansi ovisi o početnoj sposobnosti modela jer lošiji modeli iz prethodnog zadatka svojim performansama predstavljaju manji napredak u odnosu na nasumičnu klasifikaciju, pa je posljedično manja i mogućnost njihovog pogoršavanja.

6 Zaključak

U sklopu rada proučeni su veliki jezični modeli, sigurnosni aspekti njihovog korištenja i mogućnosti detekcije i maskiranja njihovih odgovora. U prvom poglavlju opisano je područje obrade prirodnog jezika i detaljno je prikazana arhitektura transformera na kojoj se temelje veliki jezični modeli. U drugom poglavlju istražena je sigurnost velikih jezičnih modela i opisani su glavni rizici i vrste napada. U trećem poglavlju opisan je postupak prikupljanja pitanja koja pokrivaju širok spektar uporabe velikih jezičnih modela i odgovora koje su odabrani veliki jezični modeli generirali za njih. U četvrtom poglavlju opisan je pristup rješavanju problema klasifikacije velikog jezičnog modela na temelju odgovora. Prikazani su rezultati dobiveni korištenjem tri različita klasifikacijska modela. U posljednjem poglavlju opisana je metoda maskiranja odgovora korištenjem parafraziranja manjim jezičnim modelom i prikazanu su rezultati klasifikacije maskiranih odgovora najboljih klasifikatora iz prijašnjeg poglavlja. Veliki jezični modeli postaju nezaobilazan alat u rješavanju i olakšavanju mnogih zadataka, a njihova sigurnost nastavlja biti važan i aktualan problem kojem je nužno posvetiti pozornost s ciljem njihove sigurne integracije.

Literatura

[1] Nerdynav, »107 Up-to-Date ChatGPT Statistics & User Numbers,« 2024. [Mrežno]. Poveznica: https://nerdynav.com/chatgpt-statistics/.

[2] A. Karpathy, »The Unreasonable Effectiveness of Recurrent Neural Networks,« 2015. [Mrežno]. Poveznica: http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

[3] A. Vaswani, N. Shazeer, N. Parmar i J. Uszkoreit, »Attention Is All You Need,« 2017. [Mrežno]. Poveznica: https://arxiv.org/abs/1706.03762.

[4] Meta, »Introducing Meta Llama 3: The most capable openly available LLM to date,« 2024. [Mrežno]. Poveznica: https://ai.meta.com/blog/meta-llama-3/.

[5] Educating Silicon, »How much LLM training data is there, in the limit?,« 2024. [Mrežno]. Poveznica: https://www.educatingsilicon.com/2024/05/09/how-much-llm-training-data-is-there-in-the-limit/.

[6] J. Wei, Y. Tay, R. Bommasani i C. Raffel, »Emergent Abilities of Large Language Models,« 2022. [Mrežno]. Poveznica: https://arxiv.org/abs/2206.07682.

[7] T. B. Brown, B. Mann, N. Ryder i M. Subbiah, »Language Models are Few-Shot Learners,« 2020. [Mrežno]. Poveznica: https://arxiv.org/abs/2005.14165.

[8] Y. Wang, H. Li i X. Han, »Do-Not-Answer: A Dataset for Evaluating Safeguards in LLMs,« 2023. [Mrežno]. Poveznica: https://arxiv.org/pdf/2308.13387.

[9] L. Lizhi, M. Honglin, Z. Zenan, W. Renxi i T. Baldwin, »Against The Achilles' Heel: A Survey on Red Teaming for,« 2024. [Mrežno]. Poveznica: https://arxiv.org/pdf/2404.00629.

[10] R. Taori, I. Gulrajani i T. Zhang, »Alpaca: A Strong, Replicable Instruction-Following Model,« 2023.. [Mrežno]. Poveznica: https://crfm.stanford.edu/2023/03/13/alpaca.html.

[11] J. Liu, L. Cui, H. Liu i D. Huang, »LogiQA: A Challenge Dataset for Machine Reading Comprehension with Logical Reasoning,« 2020.. [Mrežno]. Poveznica: https://arxiv.org/abs/2007.08124.

[12] M. Völske, M. Potthast, S. Syed i B. Stein, »TL;DR: Mining Reddit to Learn Automatic Summarization,« 2017.. [Mrežno]. Poveznica: https://aclanthology.org/W17-4508/.

[13] M. Koupaee i W. Y. Wang, »WikiHow: A Large Scale Text Summarization Dataset,« 2018.. [Mrežno]. Poveznica: https://arxiv.org/abs/1810.09305.

[14] The Stanford Natural Language Processing Group, »Stemming and lemmatization,« 2009.. [Mrežno]. Poveznica: https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html.

[15] J. Devlin, M.-W. Chang, K. Lee i K. Toutanove, »BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,« 2018.. [Mrežno]. Poveznica: https://arxiv.org/abs/1810.04805.

[16] J. Šnajder, »Jezgrene metode,« 2022.. [Mrežno]. Poveznica: https://www.fer.unizg.hr/_download/repository/SU1-2022-P10-JezgreneMetode.pdf.

[17] M. Bilenko, »Introducing Phi-3: Redefining what’s possible with SLMs,« 2024. [Mrežno]. Poveznica: https://azure.microsoft.com/en-us/blog/introducing-phi-3-redefining-whats-possible-with-slms/.

Sustav zaštite kibernetičke sigurnosti pomoću maskiranja velikog jezičnog modela

Sažetak

Veliki jezični modeli zbog svojih sposobnosti razumijevanja i generiranja teksta postaju neizostavan alat u rješavanju mnogih zadataka. Činjenica da obrađuju upite napisane prirodnim jezikom čini ih ranjivima na razne vrste napada koji iskorištavaju varijabilnost te vrste ulaznih podataka. U sklopu ovog rada istraženi su veliki jezični modeli, sigurnosni aspekti njihovog korištenja, metode klasifikacije velikih jezičnih modela na temelju njihovih odgovora i maskiranje odgovora modela s ciljem onemogućavanje točne klasifikacije.

Ključne riječi: veliki jezični modeli, transformeri, mehanizam pažnje, klasifikacija, strojno učenje, kibernetička sigurnost

Cybersecurity Protection System Using Large Language Model Masking

Abstract

Large language models are becoming an indispensable tool in solving many tasks because of their abilities of understanding and manipulating text. The fact that they process natural language prompts makes them vulnerable to many kinds of attacks which exploit the variability of this type of input data. This paper examines large language models, the security aspects of using them, methods of classifying large language models based on their responses and the masking of their responses with the aim of obstructing accurate classification.

Key words: large language models, transformers, attention mechanism, classification, machine learning, cybersecurity