Sustav za upravljanje halucinacijama velikih jezičnih modela

Vrljić, Bjanka

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 638

SUSTAV ZA UPRAVLJANJE HALUCINACIJAMA VELIKIH JEZIČNIH MODELA

Bjanka Vrljić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 638

SUSTAV ZA UPRAVLJANJE HALUCINACIJAMA VELIKIH JEZIČNIH MODELA

Bjanka Vrljić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 638

Pristupnica:	**Bjanka Vrljić (0036521774)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	prof. dr. sc. Ivica Botički

Zadatak: **Sustav za upravljanje halucinacijama velikih jezičnih modela**

Opis zadatka:

U diplomskom radu je potrebno razviti novi pristup upravljanju halucinacijama kod velikih jezičnih modela. Sustav treba biti u mogućnosti samostalno procjenjivati i poboljšavati svoje rezultate, te koristiti višestruke module za detekciju halucinacija, provjeru činjenica, evaluaciju koherentnosti i procjenu relevantnosti sadržaja. Sustav treba omogućiti automatsko prepoznavanje i klasifikaciju halucinacija u generiranim tekstovima te pružiti točnu dijagnozu o kvaliteti i pouzdanosti jezičnih modela. Pri radu, sustav treba koristiti i uspoređivati rezultate više velikih jezičnih modela, unapređujući sposobnost detekcije i smanjenja halucinacija kroz međusobno djelovanje različitih velikih jezičnih modela. Prototip sustava potrebno je ostvariti kao web-aplikaciju koja korisnicima omogućuje postavljanje upita, odabir LLM-ova koje žele uključiti te prikaz rezultata na intuitivan način.

Rok za predaju rada: 28. lipnja 2024.

SADRŽAJ

1. Uvod

Umjetna inteligencija i obrada prirodnog jezika doživjele su značajan napredak tijekom posljednjih desetljeća. Veliki jezični modeli poput GPT-3 i GPT-4 drastično su promijenili način na koji računala mogu generirati i razumjeti ljudski jezik. Ovi modeli imaju širok spektar primjena, od automatizirane korisničke podrške do stvaranja kreativnog sadržaja. Unatoč njihovim impresivnim sposobnostima, jedan od ključnih izazova je problem halucinacija, odnosno generiranja lažnih ili neutemeljenih informacija.

Halucinacije u velikim jezičnim modelima predstavljaju problem jer mogu rezultirati netočnim ili zavaravajućim sadržajem. To je posebno štetno u domenama kao što su zdravstvo, pravo i novinarstvo, gdje točnost i pouzdanost informacija imaju ključnu ulogu. Generirani tekstovi koji sadrže halucinacije mogu imati ozbiljne posljedice, od medicinskih pogrešaka do pravnih nesporazuma i dezinformacija u medijima. Stoga je detekcija halucinacija važna za osiguranje kvalitete i pouzdanosti sustava umjetne inteligencije.

Postojeći pristupi detekciji halucinacija često zahtijevaju pristup velikim vanjskim bazama podataka ili internim vjerojatnostima modela, što može biti ograničavajuće u praktičnim primjenama. Vanjske baze podataka mogu biti nedostupne ili nepotpune, a pristup internim vjerojatnostima modela često nije moguć zbog ograničenja pristupa ili zaštite intelektualnog vlasništva. To stvara potrebu za razvijanjem metoda koje mogu raditi neovisno o tim resursima.

Razvoj tehnologija za detekciju halucinacija postao je važno područje istraživanja. Jedan od značajnih pristupa je SelfCheckGPT metoda, koja omogućuje detekciju halucinacija bez potrebe za vanjskim bazama podataka ili internim vjerojatnostima modela. U ovom radu predstavljamo chatbot aplikaciju s integriranim sustavom za detekciju halucinacija, razvijenim pomoću SelfCheckGPT biblioteke. Aplikacija omogućuje korisnicima interakciju s različitim jezičnim modelima te podešavanje postavki za detekciju halucinacija.

Cilj ovog rada je detaljno opisati razvoj i implementaciju web aplikacije za detek-

ciju halucinacija, te raspraviti prednosti i ograničenja sustava. U sklopu rada objašnjeni su ključni koncepti i tehnologije korištene u ovom istraživanju. Također se razmatraju postojeći pristupi detekciji halucinacija i njihova ograničenja.

2. Teorijska pozadina

2.1. Obrada prirodnog jezika

Obrada prirodnog jezika (engl. *natural language processing, NLP*) je područje umjetne inteligencije (engl. *artificial intelligence, AI*) koje se fokusira na interakciju između računala i ljudi putem prirodnog (ljudskog) jezika. [8] Krajnji cilj NLP-a je omogućiti računalu da razumije, interpretira i generira prirodni jezik na način koji je smislen i koristan. NLP je interdisciplinarno područje koje kombinira tehnike iz lingvistike i strojnog učenja (engl. *machine learning*), što omogućuje obradu i analizu velikih količina podataka prirodnog jezika.

Počeci NLP-a sežu do 1950-ih godina kada je Alan Turing predložio Turingov test kao kriterij za umjetnu inteligenciju. Rani napreci u NLP-u uključivali su sustave temeljene na pravilima (engl. *rule-based systems*) koji su se uvelike oslanjali na ručno izrađena gramatička pravila. Ti sustavi su s vremenom evoluirali i počeli uključivati statističke metode i tehnike strojnog učenja, što je dovelo do značajnih napredaka u ovom području. [21]

NLP obuhvaća širok raspon zadataka. Označavanje dijelova govora (engl. *part-of-speech tagging*) dodjeljuje gramatičku funkciju (npr. imenica, glagol, pridjev) svakoj riječi unutar rečenice, pružajući dublje razumijevanje strukture rečenice. Prepoznavanje imenovanih entiteta (engl. *named entity recognition*) fokusira se na izdvajanje i klasificiranje specifičnih entiteta spomenutih u tekstu, kao što su ljudi, organizacije i lokacije. Prepoznavanje govora (engl. *speech recognition*) omogućuje računalu da prepozna i interpretira ljudski govor te ga pretvori u tekst. Generiranje teksta (engl. *text generation*) omogućuje računalu da stvara tekstualne sadržaje, od kreativnog pisanja do strožih formata poput koda ili skripti. Strojno prevođenje (engl. *machine translation*) omogućava automatsko prevođenje teksta s jednog jezika na drugi. Sažimanje teksta (engl. *text summarization*) sažima duge tekstove u kraće sažetke, zadržavajući osnovno značenje bez žrtvovanja ključnih informacija. Analiza sentimenta (engl. *sentiment analysis*) je proces određivanja sentimenta ili emocije izražene u tekstu, identi-

ficirajući emocionalni ton ili mišljenje koje je izrazio autor. [8]

Primjene NLP-a su brojne i raznovrsne. Tražilice (engl. *search engines*) koriste NLP za razumijevanje korisničkih upita i pružanje relevantnih rezultata pretraživanja, koristeći analizu konteksta i namjere upita. Sustavi za prepoznavanje govora koriste NLP za pretvaranje govora u tekst, te se često primjenjuju u virtualnim asistentima i za transkripciju. Sažimanje teksta uključuje generiranje sažetka većeg teksta, što je korisno za agregaciju vijesti i sažimanje dokumenata. Chatbotovi koriste NLP za razumijevanje i odgovaranje na korisničke upite na konverzacijski način, a široko se koriste u službi za korisnike, virtualnim asistentima i sustavima za pretraživanje informacija. Analiza sentimenta na društvenim mrežama koristi se za analizu sentimenta izraženog u objavama, komentarima i recenzijama, pomažući tvrtkama u razumijevanju mišljenja korisnika i poboljšanju svojih proizvoda i usluga.

Međutim, postoje i brojni izazovi u NLP-u. Prirodni jezik je inherentno dvosmislen, s riječima i rečenicama koje često imaju više značenja, što predstavlja značajan izazov za NLP sustave. Razumijevanje konteksta u kojem je riječ ili fraza korištena ključno je za točno procesiranje jezika, što zahtijeva sofisticirane modele koji mogu obuhvatiti kontekstualne informacije. Razvoj NLP sustava koji mogu obrađivati više jezika također je složen zadatak zbog razlika u gramatici, sintaksi i semantici među jezicima. Nedostatak označenih podataka i jezičnih resursa za mnoge jezike dodatno otežava razvoj preciznih NLP modela za te jezike.

2.2. Jezični modeli

Jezični modeli (engl. *language models, LMs*) su modeli strojnog učenja koji omogućuju računalima razumijevanje, generiranje i obradu ljudskog jezika. Oni su temeljni alat u obradi prirodnog jezika i koriste se za široki spektar primjena, uključujući prevođenje, sažimanje teksta, prepoznavanje govora, chatbotove itd. Razvoj jezičnih modela omogućio je značajan napredak u sposobnosti računala da interpretiraju i generiraju prirodni jezik.

Tradicionalni jezični modeli temeljili su se na pravilima i statističkim metodama. Pristupi temeljeni na pravilima koristili su unaprijed definirane gramatičke i sintaktičke strukture kako bi analizirali i generirali jezik. Iako su ti modeli bili korisni za jednostavnije zadatke, imali su ograničenu sposobnost razumijevanja složenih jezičnih struktura i konteksta. S druge strane, statistički jezični modeli koristili su vjerojatnosti za predviđanje slijeda riječi u tekstu. Ovi modeli, poput n-gram modela, procjenjivali su vjerojatnost pojave riječi na temelju prethodnih n riječi u nizu. Iako su statistički

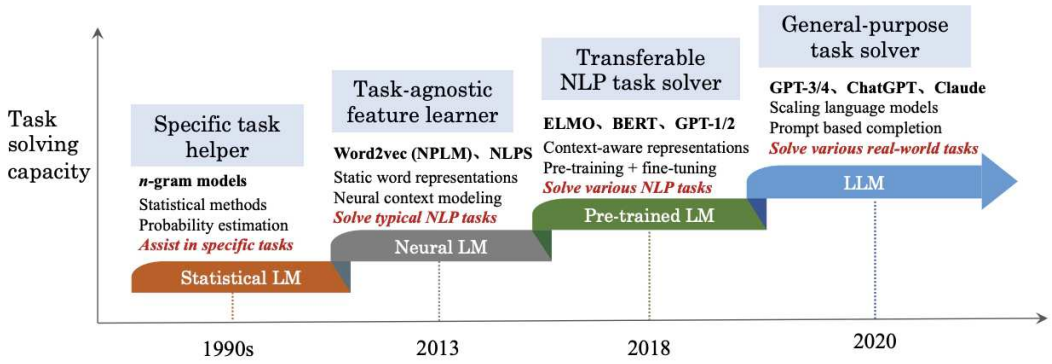
modeli bili učinkovitiji od pristupa temeljenih na pravilima, i dalje su imali ograničenja u prepoznavanju dugoročnih ovisnosti (engl. *long-term dependencies*) i složenih jezičnih struktura. [20] [30] [31]

S razvojem strojnog učenja, posebno dubokog učenja (engl. *deep learning*), jezični modeli doživjeli su značajan napredak. Duboko učenje koristi složene neuronske mreže koje mogu samostalno naučiti reprezentacije jezika iz velikih količina podataka. Neuronske mreže složenijih arhitektura, poput rekurentnih neuronskih mreža (engl. *recurrent neural network, RNN*) i mreža dugog kratkoročnog pamćenja (engl. *long short-term memory, LSTM*), omogućuju modelima da zadrže informacije kroz dulje vremenske periode i bolje prepoznaju složene jezične obrasce (engl. *language patterns*).

2.3. Veliki jezični modeli

Veliki jezični modeli (engl. *large language models, LLMs*) predstavljaju značajan napredak u području obrade prirodnog jezika. Korištenjem ogromnih količina podataka i sofisticiranih arhitektura, LLM-ovi su transformirali brojne aplikacije, od chatbotova i virtualnih asistenata do automatiziranog stvaranja sadržaja i usluga prevođenja.

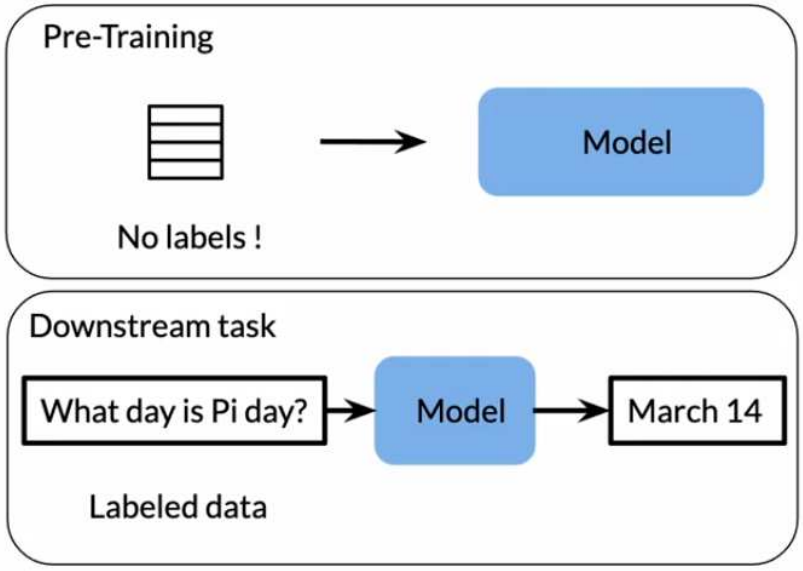
Jedan od najvažnijih napredaka za LLM-ove je razvoj transformer arhitekture [39]. Transformeri koriste mehanizam samopozornosti (engl. *self-attention mechanism*), koji omogućuje modelima da obrate pažnju na različite dijelove ulaznog teksta istovremeno te procjene važnost različitih riječi u rečenici. Ovaj pristup omogućuje učinkovitu obradu velikih količina teksta i prepoznavanje dugoročnih ovisnosti. Modeli temeljeni na transformerima, kao što su BERT (Bidirectional Encoder Representations from Transformers) [19] i GPT (Generative Pre-trained Transformer) [32], postali su standard za mnoge NLP zadatke.



Slika 2.1: Razvoj jezičnih modela. [41]

BERT je dvosmjerni model koji uzima u obzir kontekst riječi s obje strane ciljne riječi, što omogućuje bolje razumijevanje značenja riječi u kontekstu. Ovaj model posebno je učinkovit u zadacima razumijevanja jezika, poput prepoznavanja entiteta, analize sentimenta i odgovaranja na pitanja (engl. *question answering*). GPT je generativni model koji je treniran da predvidi sljedeću riječ u sekvenci, uzimajući u obzir kontekst, odnosno riječi koje su joj prethodile. Ovaj model je posebno učinkovit u zadacima generiranja teksta, kao što su pisanje eseja, generiranje dijaloga i kreativno pisanje.

Proces treniranja LLM-ova uključuje dvije glavne faze: predtreniranje (engl. *pre-training*) i fino podešavanje (engl. *fine-tuning*). Tijekom predtreniranja, model se trenira na velikim količinama neoznačenog teksta poput Wikipedije. Na taj način model nauči reprezentacije riječi i rečenica te temeljne semantičke odnose s kojima su povezani. Ova faza omogućuje modelu razvijanje općeg razumijevanja jezika. S druge strane, fino podešavanje uključuje treniranje modela na manjem, specifičnom skupu podataka za određene zadatke, prilagođavajući ga za posebne primjene. Ovaj dvofazni proces omogućuje LLM-ovima postizanje visokih performansu u raznim NLP zadacima.



Slika 2.2: Predtreniranje i fino podešavanje. [37]

LLM-ovi imaju široku primjenu u mnogim industrijama i svakodnevnim tehnologijama. U službi za korisnike, chatbotovi bazirani na LLM-ovima mogu obrađivati složene upite i pružati točne odgovore, poboljšavajući korisničko iskustvo i učinkovitost. U stvaranju sadržaja, LLM-ovi mogu generirati članke, izvješća, pa čak i kreativne

tekstove, pomažući piscima i automatizirajući repetitivne administrativne zadatke. U zdravstvenom sektoru, ovi modeli mogu pomoći u analizi medicinskih zapisa, ekstrakciji relevantnih informacija i čak u pružanju preliminarnih dijagnoza.

Unatoč njihovim prednostima, jezični modeli suočavaju se s izazovima kao što su pristranost (engl. *bias*) podataka, etička pitanja i potreba za velikim računalnim resursima. Pristranost podataka može dovesti do nepoželjnih i netočnih rezultata, dok etička pitanja uključuju privatnost podataka i potencijalnu zloupotrebu tehnologije. Nadalje, treniranje i implementacija velikih jezičnih modela zahtijeva značajne računalne resurse, što može biti prepreka za mnoge organizacije.

2.4. Chatbot

Chatbot je računalni program dizajniran za simulaciju razgovora s korisnicima [6], najčešće putem web aplikacije. Koristi tehnike obrade prirodnog jezika kako bi razumio i odgovorio na tekstualne ili glasovne upite korisnika. Chatbotovi su postali ključna komponenta u mnogim industrijama, uključujući korisničku podršku, e-trgovinu, obrazovanje i zdravstvo, omogućujući organizacijama da poboljšaju korisničko iskustvo i optimiziraju učinkovitost.

Razvoj chatbotova započeo je sredinom 20. stoljeća s jednostavnim programima poput ELIZA, koju je stvorio Joseph Weizenbaum 1966. godine. ELIZA je koristila prepoznavanje ključnih riječi i jednostavne predloške odgovora kako bi simulirala razgovor. [20] Iako su rani chatbotovi bili ograničeni u svojoj sposobnosti da razumiju i generiraju složene odgovore, postavili su temelje za daljnji razvoj.

S napretkom NLP-a i strojnog učenja, moderni chatbotovi postali su mnogo sofisticiraniji. Koriste različite algoritme strojnog učenja i dubokog učenja kako bi analizirali prirodni jezik, prepoznali namjeru korisnika i generirali odgovarajuće odgovore. Dva ključna pristupa u razvoju chatbotova su pristup temeljen na pravilima i pristup temeljen na strojnom učenju. [18]

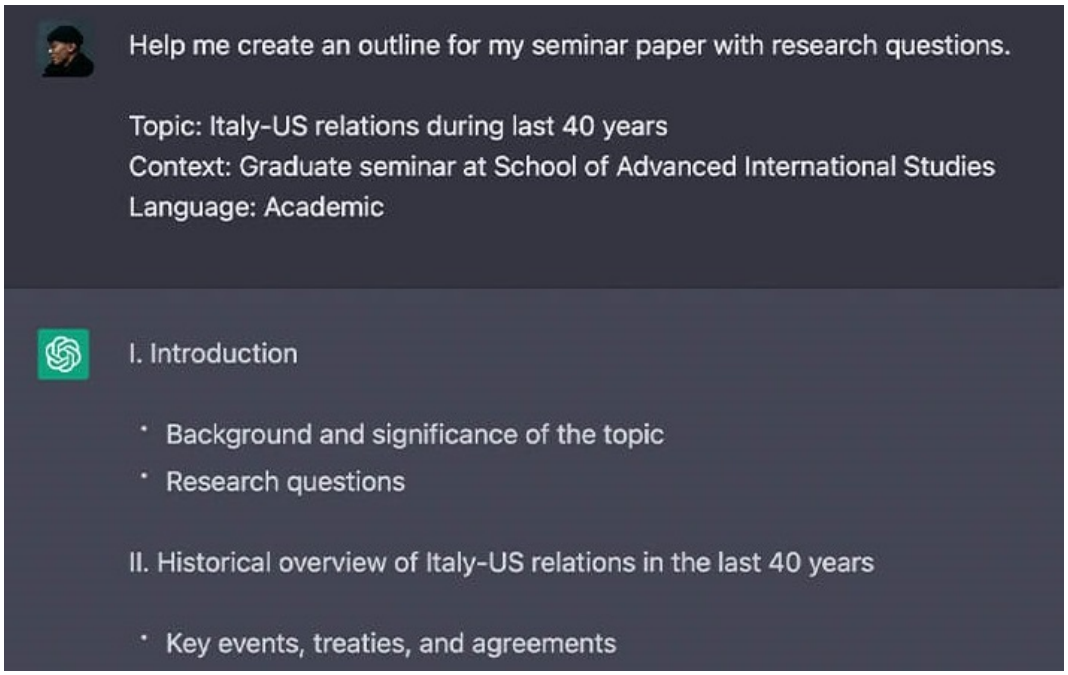
Chatbotovi temeljeni na pravilima oslanjaju se na unaprijed definirana pravila i obrasce kako bi generirali odgovore. Ovi chatbotovi koriste skripte i stabla odluke (engl. *decision trees*) kako bi vodili korisnika kroz unaprijed određene scenarije. Iako su učinkoviti za jednostavne i specifične zadatke, ograničeni su u svojoj sposobnosti da se prilagode složenijim ili nepredviđenim upitima.

S druge strane, chatbotovi temeljeni na strojnom učenju koriste tehnike strojnog učenja i dubokog učenja kako bi automatski učili iz velikih skupova podataka. Modeli poput rekurentnih neuronskih mreža (RNN) i transformera, omogućuju chatboto-

vima da razumiju kontekst i generiraju prirodnije i raznovrsnije odgovore. Na primjer, GPT-3, LLM kojeg je razvio OpenAI, koristi transformer arhitekturu kako bi generirao koherentan i kontekstualno relevantan tekst, što ga čini izuzetno moćnim alatom za chatbotove.

Jedan od glavnih izazova u razvoju chatbotova je razumijevanje namjere korisnika. Prepoznavanje namjere uključuje identifikaciju svrhe ili cilja korisnikove poruke, što je ključno za generiranje odgovarajućeg odgovora. Ovaj proces često uključuje klasifikaciju namjere pomoću algoritama strojnog učenja treniranih na označenim podacima. Uz prepoznavanje namjere, chatbotovi moraju biti sposobni za rukovanje entitetima kao što su imena, datumi i lokacije, kako bi pružili točne i relevantne informacije.

Primjene chatbotova su raznovrsne i obuhvaćaju mnoge industrije. U korisničkoj podršci, chatbotovi mogu automatizirati odgovore na često postavljana pitanja, riješiti jednostavne probleme i preusmjeriti složenije upite prema ljudskim agentima. U e-trgovini, chatbotovi mogu pomagati korisnicima tijekom kupovine, preporučivati proizvode i pružati podršku u vezi narudžbi. U obrazovanju, chatbotovi mogu pružati podršku studentima, odgovarati na pitanja u vezi gradiva i pomagati u organizaciji obrazovnih materijala. U zdravstvu, chatbotovi mogu pružati informacije o simptomima, zakazivati termine i podsjećati pacijente na uzimanje lijekova. [26]



Slika 2.3: Primjer korištenja chatbota (ChatGPT). [15]

Unatoč njihovim prednostima, chatbotovi se suočavaju s izazovima kao što su razumijevanje složenih upita, održavanje prirodnog toka razgovora i izbjegavanje pris-

tranosti. Razumijevanje složenih i višeznačnih upita zahtijeva napredne algoritme koji mogu analizirati kontekst i generirati točne odgovore. Održavanje prirodnog toka razgovora uključuje sposobnost prepoznavanja i odgovaranja na nepredviđene promjene u razgovoru. Pristranost može nastati iz podataka na kojima su chatbotovi trenirani, što može dovesti do nepoželjnih i netočnih odgovora. Rješavanje ovih izazova zahtijeva kontinuirano istraživanje i razvoj te etička razmatranja.

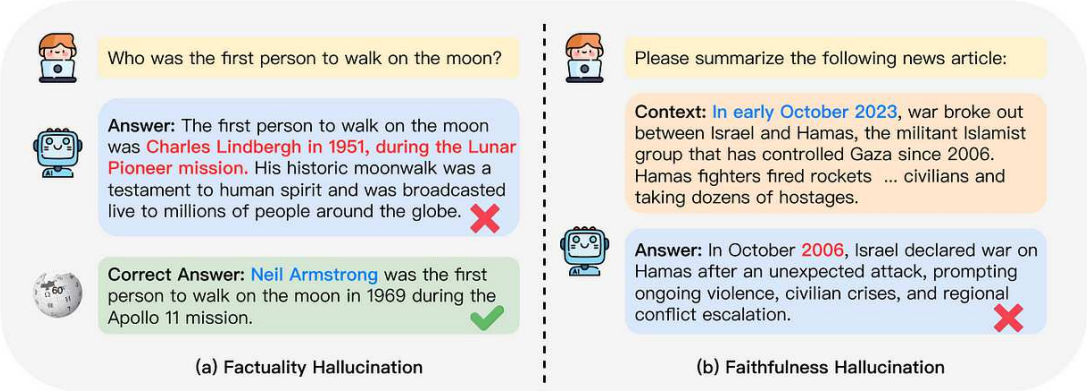
2.5. Halucinacije

U kontekstu umjetne inteligencije i obrade prirodnog jezika, halucinacije se odnose na pojavu kada modeli generativne umjetne inteligencije (engl. *generative AI*), poput velikih jezičnih modela, stvaraju lažne ili neutemeljene informacije koje nisu prisutne u ulaznim podacima. [7] Ove halucinacije mogu biti problematične jer generirani sadržaj može biti netočan, zavaravajuć ili potencijalno štetan.

Halucinacije u NLP modelima mogu se pojaviti iz nekoliko razloga. Modeli su trenirani na velikim skupovima podataka koji sadrže različite vrste informacija, uključujući i one koje su netočne ili neprovjerene. Budući da modeli nemaju inherentno razumijevanje istine, mogu generirati informacije koje su gramatički i kontekstualno točne, ali činjenično netočne. Također, modeli mogu stvarati halucinacije zbog nesigurnosti ili dvosmislenosti u ulaznim podacima. Kada model primi nedovoljno specifične ili nejasne upite, može pokušati "popuniti praznine" stvaranjem uvjerljivog, ali izmišljenog sadržaja. [34]

Različite vrste halucinacija mogu se pojaviti u LLM-ovima. Jedna vrsta su činjenične halucinacije (engl. *factuality hallucinations*), gdje model generira lažne činjenice, datume, imena ili događaje. Na primjer, chatbot može odgovoriti na pitanje o povijesnom događaju stvaranjem netočnih informacija koje zvuče uvjerljivo. Druga vrsta su halucinacije po vjernosti (engl. *faithfulness hallucinations*), gdje model stvara odgovore koji nisu povezani ili nisu konzistentni s danim kontekstom ili temom razgovora. Ove halucinacije mogu narušiti korisničko iskustvo i dovesti do pogrešnih zaključaka. [22]

Razvoj tehnika za smanjenje halucinacija u LLM-ovima aktivno je područje istraživanja. Jedan od mogućih pristupa je poboljšanje kvalitete i raznolikosti skupa podataka na kojem se modeli treniraju. Korištenje pouzdanih i verificiranih izvora informacija može smanjiti rizik od halucinacija. Drugi pristup uključuje korištenje metoda post-procesiranja za provjeru činjenica i verifikaciju generiranog sadržaja. Ovi sustavi mogu automatski uspoređivati generirane odgovore s bazama podataka ili referentnim



Slika 2.4: Primjer dviju vrsta halucinacija. [22]

izvorima kako bi identificirali i ispravili netočne informacije.

Jedan od izazovnih aspekata halucinacija je njihova detekcija. Budući da halucinacije mogu biti suptilne i uvjerljive, njihovo prepoznavanje zahtijeva sofisticirane algoritme koji mogu analizirati semantički sadržaj i kontekst generiranog teksta. Razvoj naprednih metoda za detekciju halucinacija uključuje korištenje dubokog učenja, provjere činjenica i tehnika razumijevanja jezika kako bi se identificirali potencijalni problemi u generiranom sadržaju.

Halucinacije u LLM-ovima također dovode važna etička pitanja u obzir. Generiranje netočnih ili zavaravajućih informacija može imati ozbiljne posljedice, posebno u kontekstima gdje se točnost i pouzdanost informacija smatraju kritičnima, poput medicine, prava ili novinarstva. Potrebno je razviti etičke smjernice i regulatorne okvire koji će osigurati odgovornu upotrebu ovih tehnologija i minimizirati rizik od štetnih učinaka.

Unatoč izazovima, LLM-ovi imaju potencijal za značajnu korist u različitim domenama. Na primjer, mogu se koristiti za generiranje sadržaja, podršku korisnicima i prevođenje jezika. Kako bi se maksimizirale ove koristi i minimizirale halucinacije, ključno je nastaviti s istraživanjima i razvojem kako bi se poboljšala točnost i pouzdanost ovih modela. Razumijevanje uzroka halucinacija i razvoj učinkovitih metoda za njihovo smanjenje ključni su za unapređenje kvalitete i pouzdanosti modela obrade prirodnog jezika. Kroz kontinuirano istraživanje i primjenu etičkih smjernica, nastoji se postići ravnoteža između inovacije i odgovornosti, omogućujući sigurnu i korisnu integraciju ovih tehnologija u razne aspekte društva.

Uz tehnološke pristupe, edukacija korisnika o mogućnostima i ograničenjima LLM-ova ima ključnu ulogu u postavljanju očekivanja i smanjenju rizika od pogrešne upotrebe. Transparentnost s obzirom na način rada modela, izvore podataka i proces treni-

ranja može pomoći korisnicima da bolje razumiju potencijalne slabosti i prednosti ovih sustava. Osim toga, suradnja između istraživača, inženjera i regulatornih tijela ključna je za izgradnju povjerenja i osiguranje da LLM-ovi donose korist uz minimalne negativne učinke.

3. Srodni radovi

Prethodni radovi u području nude sveobuhvatan pregled halucinacija prisutnih pri generiranju prirodnog jezika (engl. *natural language generation, NLG*) [24], analizirajući fenomen halucinacija u različitim zadacima NLG-a kao što su apstraktno sažimanje teksta (engl. *abstractive summarization*), generiranje dijaloga, generativno odgovaranje na pitanja, generiranje teksta iz podataka (engl. *data-to-text generation*) i strojno prevođenje (engl. *machine translation*). Istraživanja svrstavaju halucinacije u kategorije, identificiraju njihove uzroke te proučavaju razne metode za njihovo otkrivanje i smanjenje. Intrinzične halucinacije uključuju pogreške unutar samog generiranog teksta, poput logičkih nekonzistentnosti i gramatičkih pogrešaka. Ekstrinzične halucinacije zahtijevaju vanjsko znanje kako bi se identificirale pogreške, poput pogrešnih činjenica, datuma, imena ili događaja. [24]

Huang i suradnici [23] istraživali su halucinacije pri sažimanju teksta, ističući kako generirani sažeci često sadrže pogrešne činjenice i nekonzistentnosti u odnosu na izvorni tekst. Njihovo istraživanje naglašava važnost održavanja činjenične točnosti u sažecima kako bi se osigurala pouzdanost i primjenjivost generiranih rezultata. Shuster i suradnici [35] fokusirali su svoje istraživanje na generiranje dijaloga, istražujući kako halucinacije mogu utjecati na koherentnost i prirodnost generiranih razgovora. Njihov rad ističe izazove u održavanju kontekstualne točnosti tijekom generiranja dijaloga, što je ključno za interaktivne sustave umjetne inteligencije.

Prethodna istraživanja u području identificiraju nekoliko čimbenika koji doprinose halucinacijama, uključujući kvalitetu podataka za treniranje, arhitekturu modela i dizajn upita. Loša kvaliteta podataka ili šumoviti podaci (engl. *noisy data*) za treniranje mogu dovesti do generiranja netočnih informacija, dok određene arhitekture modela mogu biti sklonije generiranju halucinacija zbog svoje složenosti i načina obrade informacija. Dizajn upita također igra ključnu ulogu jer loše strukturirani upiti često rezultiraju netočnim izlazima.

Za detekciju halucinacija koriste se razne tehnike, kao što su moduli za provjeru činjenica koji uspoređuju generirani sadržaj s provjerenim bazama podataka, te pro-

vjera konzistentnosti kako bi se osigurala logička koherentnost unutar teksta. Također se koriste ljudske procjene gdje ljudski ocjenjivači ručno pregledavaju generirani sadržaj kako bi identificirali i klasificirali halucinacije. Ove metode pružaju sveobuhvatan pristup identificiranju halucinacija i uspostavljanju temeljne istine (engl. *ground truth*) za evaluaciju modela. Liu i suradnici [28] razvili su novi skup podataka za detekciju halucinacija u generiranim tekstovima manipuliranjem činjenično točnih tekstova kako bi simulirali halucinacije. Njihov pristup pruža važan resurs za procjenu i poboljšanje robusnosti modela u prepoznavanju netočnih ili nepodržanih informacija.

Strategije smanjenja halucinacija opisane u prethodnim radovima [38] uključuju generiranje uz pomoć pretraživanja (engl. *retrieval-augmented generation, RAG*), koje poboljšava točnost izlaza LLM-ova dohvaćanjem relevantnih informacija iz vanjskih baza znanja tijekom generiranja teksta. To osigurava da je sadržaj relevantan, aktualan i provjerljiv. Druge strategije uključuju inženjerstvo upita (engl. *prompt engineering*), filtre za post-procesiranje (engl. *post-processing filters*) koji automatski označavaju i ispravljaju halucinirani sadržaj, te fino podešavanje modela korištenjem visokokvalitetnih, označenih skupova podataka namijenjenih specifično za smanjenje halucinacija.

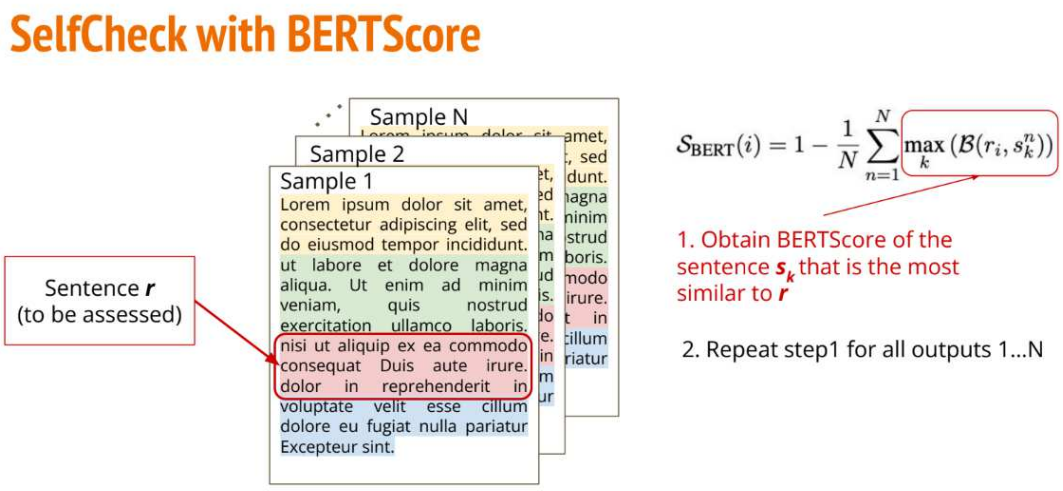
Wang i suradnici [40] predstavili su koncept samo-konzistentnosti (engl. *self consistency*) kako bi poboljšali performanse modela u složenim zadacima rasuđivanja pomoću lanca misli (engl. *chain-of-thought reasoning*). Ovaj pristup potiče modele da održavaju logičku koherentnost tijekom generiranja teksta, čime se smanjuje vjerojatnost halucinacija i poboljšava kvaliteta generiranih rezultata kod rasuđivanja u više koraka. Kadavath i suradnici [25] istraživali su tehnike samoprocjene (engl. *self-evaluation*) gdje jezični modeli procjenjuju istinitost svojih vlastitih izlaza. Ovaj pristup potiče samosvijest (engl. *self-awareness*) i samoispravljanje (engl. *self-correction*) unutar modela, potičući ih da bolje procijene vjerojatnost da su njihovi odgovori točni. Takve tehnike poboljšavaju pouzdanost generiranih tekstova u praktičnim primjenama.

3.1. SelfCheckGPT

Jedan od značajnih napredaka u području detekcije halucinacija kod velikih jezičnih modela je razvoj metode SelfCheckGPT [29]. Ova metoda nudi pristup koji ne zahtijeva vanjske baze podataka, što je čini pristupom bez resursa (engl. *zero-resource*) i metodom crne kutije (engl. *black-box*). Temeljna ideja SelfCheckGPT-a je da jezični model koji uistinu razumije neki koncept će proizvoditi konzistentne i činjenično točne odgovore na isti upit (engl. *prompt*). Suprotno tome, halucinirani sadržaj obično generira nekonzistentne i kontradiktorne odgovore.

SelfCheckGPT ima nekoliko varijanti koje koriste različite tehnike za detekciju halucinacija, osiguravajući sveobuhvatnu evaluaciju i pouzdanost izlaza LLM-a. Ove tehnike uključuju: BERTScore, odgovaranje na pitanja (engl. *question answering, QA*), n-gram, zaključivanje prirodnog jezika (engl. *natural language inference, NLI*) i LLM Prompt.

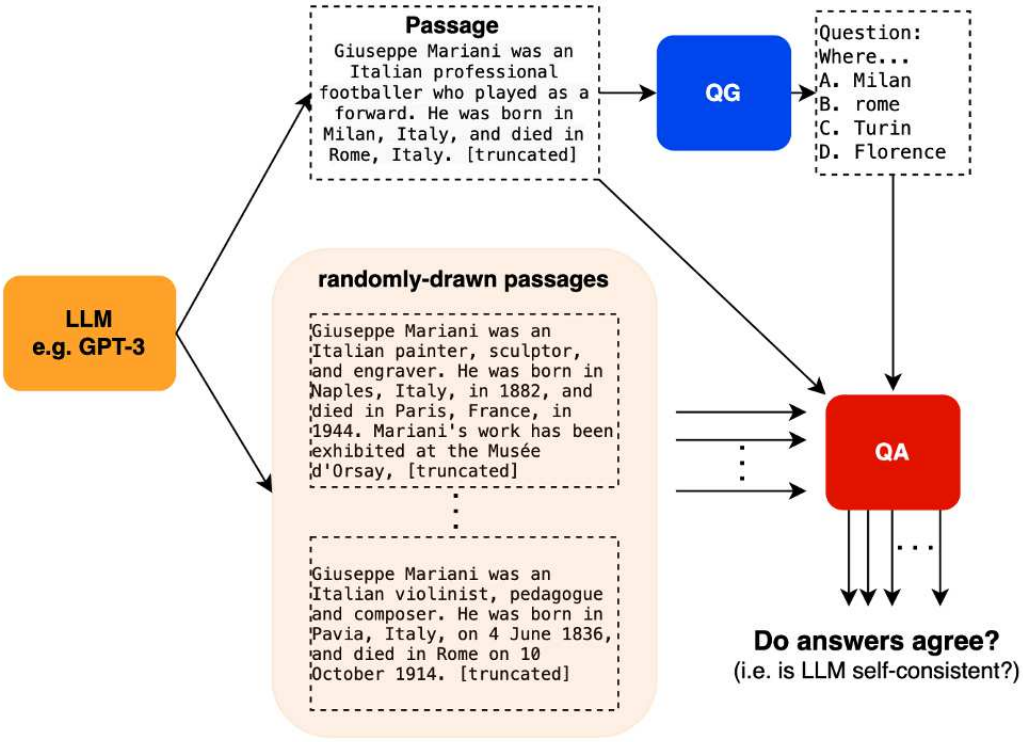
BERTScore metoda izračunava sličnost između generiranih rečenica pomoću BERTScore metrike. Visoka sličnost među više uzorkovanih odgovora, tj. uzoraka (engl. *samples*) ukazuje na činjeničnu konzistentnost (engl. *factual consistency*), dok niska sličnost sugerira potencijalne halucinacije. Ova metoda koristi RoBERTa-Large kao okosnicu (engl. *backbone*) za BERTScore kako bi izmjerila prosječan BERTScore svake rečenice iz glavnog odgovora (engl. *main response*) s najsličnijom rečenicom iz svakog uzorka. Ova tehnika je posebno učinkovita u identificiranju rečenica koje odstupaju od norme, ističući nepodudarnosti koje mogu ukazivati na halucinirani sadržaj.



Slika 3.1: SelfCheckGPT s BERTScore. [14]

QA metoda generira pitanja višestrukog izbora (engl. *multiple-choice questions*) iz glavnog odgovora i koristi neovisni QA sustav za odgovaranje na ta pitanja na temelju uzoraka. Konzistentni odgovori među uzorcima ukazuju na činjeničnu točnost, dok razlike ističu potencijalne halucinacije. Ova metoda uključuje dvije faze: generiranje pitanja i odgovaranje na pitanja. Konzistentnost odgovora temelji se na broju podudaranja i nepodudaranja između različitih odgovora na pitanja.

N-gram metoda koristi vjerojatnosti za mjerenje izglednosti (engl. *likelihood*) generiranih rečenica. Visoke log-vjerojatnosti (engl. *log-probabilities*) ukazuju na manju šansu za halucinaciju. Ova tehnika uključuje treniranje jednostavnog n-gram modela koristeći uzorke i glavni odgovor, zatim izračunavanje prosječnih log-vjerojatnosti

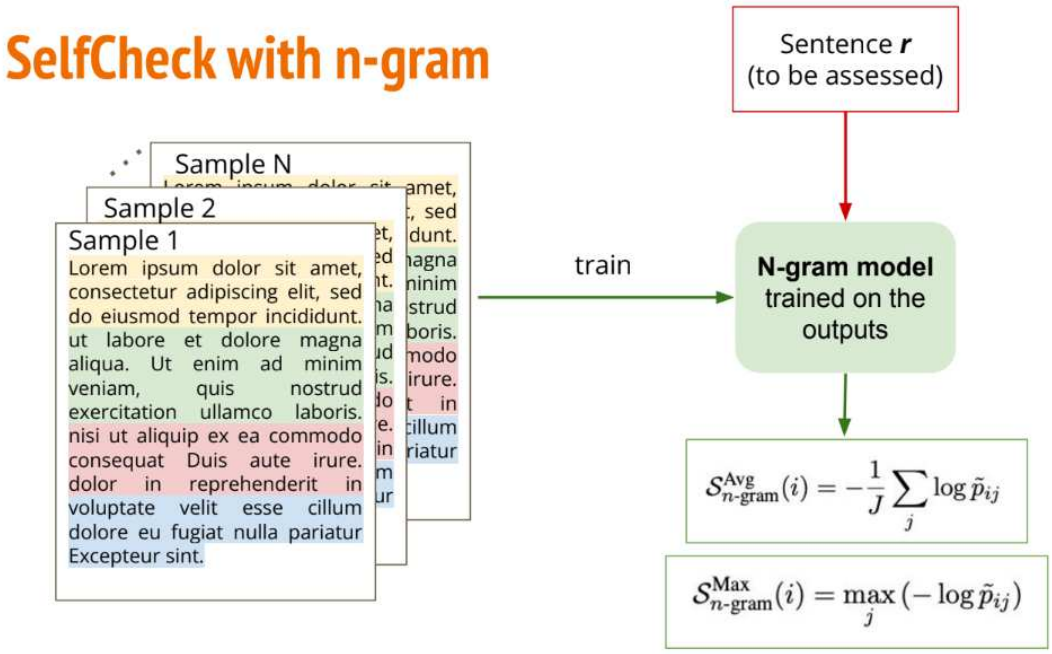


Slika 3.2: SelfCheckGPT s QA. [14]

svake rečenice u odgovoru. Takav pristup pruža statističku mjeru vjerojatnosti rečenice, što pomaže u identificiranju rečenica koje su statistički malo vjerojatne i stoga potencijalno halucinirane.

NLI određuje proizlazi li rečenica logično iz konteksta koji pružaju uzorci. Koristi ocjene implikacije i kontradikcije (engl. *entailment and contradiction scores*) za procjenu činjenične konzistentnosti. Ovaj pristup koristi DeBERTa-v3-large model, fino podešen za MNLI (Multi-Genre NLI), te normalizira vjerojatnost implikacije ili kontradikcije, fokusirajući se na ocjenu kontradikcije kako bi se istaknula nepodudaranja. Posebno je koristan za provjeru logičke konzistentnosti odgovora u zadanom kontekstu.

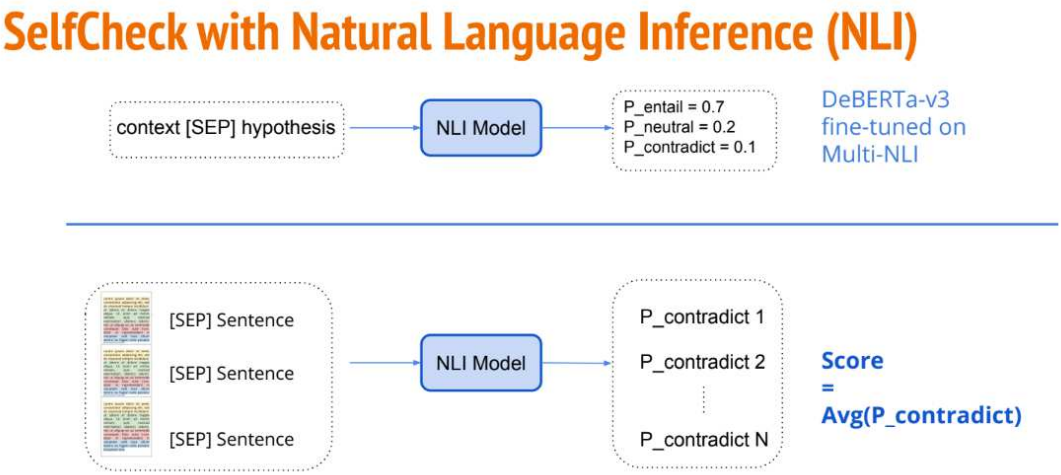
LLM Prompt uključuje izravno postavljanje upita (engl. *querying, prompting*) LLM-u kako bi mogli procijeniti je li rečenica podržana kontekstom iz uzoraka. Za svaku rečenicu iz glavnog odgovora i za svaki uzorak, pitamo LLM da li kontekst iz uzorka podupire rečenicu. Odgovori na taj upit se ocjenjuju pa uprosječe kako bi se odredila vjerojatnost halucinacije. Ova tehnika može se prilagoditi raznim LLM-ovima dostupnim na platformama poput Hugging Face, te omogućava prilagodbu upita kako bi se zadovoljili specifični zahtjevi.



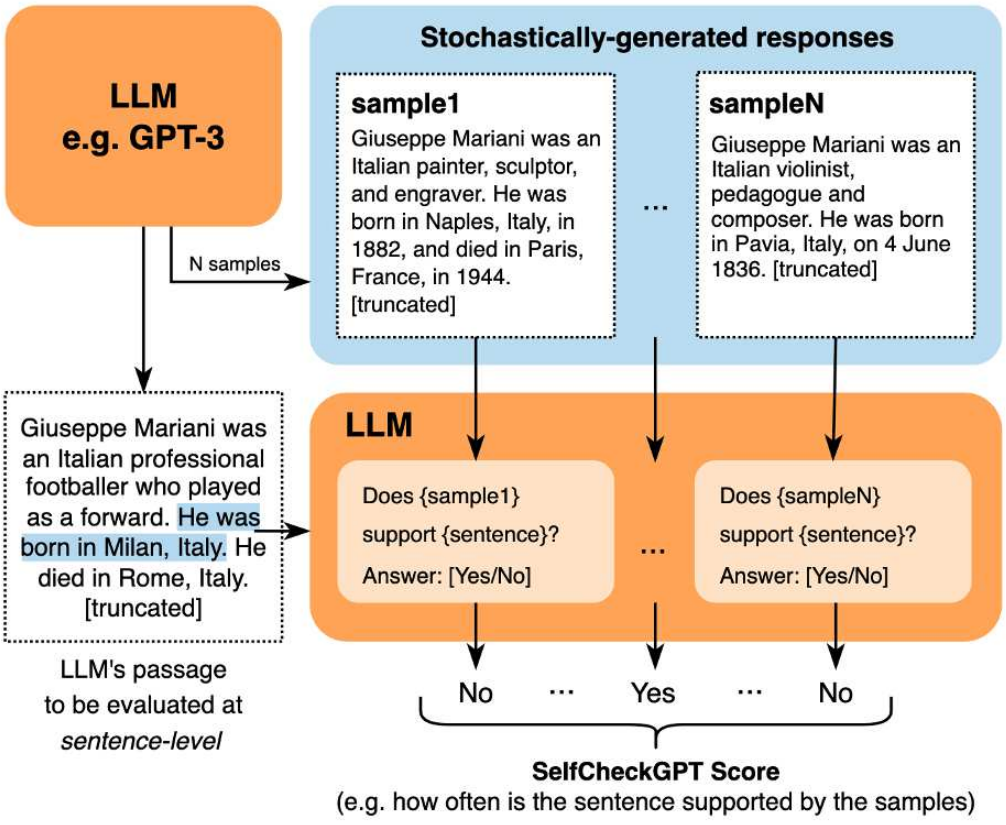
Slika 3.3: SelfCheckGPT s n-gramom. [14]

Učinkovitost SelfCheckGPT-a u identifikaciji halucinacija temelji se na njegovoj sposobnosti da uspoređuje više uzorkovanih odgovora i mjeri njihovu konzistentnost. Korištenjem različitih metoda evaluacije, SelfCheckGPT pruža robustan pristup osiguravanju pouzdanosti sadržaja generiranog umjetnom inteligencijom. Ovo je posebno važno za primjene u osjetljivim domenama kao što su zdravstvo, pravo i novinarstvo, gdje su točnost i pouzdanost informacija od ključne važnosti. Osim toga, SelfCheckGPT je pristup bez resursa, što ga čini lako prilagodljivim i skalabilnim, omogućujući integraciju u razne sustave bez potrebe za opsežnim vanjskim bazama podataka.

Razvoj i implementacija SelfCheckGPT-a imaju značajne implikacije za budućnost umjetne inteligencije i obrade prirodnog jezika. Kako jezični modeli nastavljaju evoluirati i postaju sofisticiraniji, sposobnost detekcije i smanjenja halucinacija bit će ključna za održavanje povjerenja i pouzdanosti AI sustava. SelfCheckGPT postavlja temelj za buduća istraživanja i razvoj u ovom području, naglašavajući važnost konzistentnog i točnog sadržaja generiranog umjetnom inteligencijom.



Slika 3.4: SelfCheckGPT s NLI. [14]



Slika 3.5: SelfCheckGPT s LLM Prompt. [14]

4. Metodologija

4.1. Korištene tehnologije i resursi

4.1.1. Hugging Face

Hugging Face [1] je vodeća platforma za obradu prirodnog jezika koja pruža alate i resurse za izgradnju, treniranje i primjenu modela dubokog učenja. Osnovana 2016. godine, Hugging Face platforma postala je ključna za napretke u NLP-u zahvaljujući svom opsežnom katalogu predtreniranih modela, uključujući BERT, GPT-3, T5 i mnoge druge.

Jedna od ključnih značajki Hugging Face-a je Python biblioteka Transformers, koja omogućuje jednostavnu integraciju i korištenje različitih modela dubokog učenja. Ova biblioteka podržava razne modele za zadatke poput klasifikacije teksta, generiranja teksta, strojnog prevođenja, odgovaranja na pitanja, te mnoge druge. Biblioteka je dizajnirana da bude jednostavna za korištenje, s intuitivnim programskim sučeljem (engl. *application programming interface, API*) koje omogućuje brzu implementaciju modela za mnoge primjene.

Hugging Face također nudi platformu za dijeljenje i suradnju na modelima i skupovima podataka. Korisnici mogu učitati vlastite modele i skupove podataka, čineći ih dostupnima zajednici. Ova platforma olakšava istraživanje i razvoj u NLP-u, jer istraživači i inženjeri mogu lako pristupiti najnovijim modelima i resursima koje su podijelili drugi korisnici. Hugging Face Hub je centralno mjesto za ove resurse, omogućujući jednostavno pretraživanje i preuzimanje modela i skupova podataka. Jedan od najznačajnijih doprinosa Hugging Face-a je zajednica koja se razvila oko platforme. Aktivni korisnici redovito dijele svoja saznanja, modele i rješenja. Ova zajednica doprinosi stalnom poboljšanju i inovacijama u NLP-u, jer korisnici mogu brzo usvojiti nove tehnike i pristupe koji se razvijaju.

Hugging Face također podržava integraciju s popularnim okruženjima za treniranje modela, kao što su PyTorch i TensorFlow. Ovo omogućuje korisnicima da koriste alate

i okvire s kojima su već upoznati, čineći proces treniranja i implementacije modela bržim i učinkovitijim. Hugging Face sadrži opsežnu dokumentaciju i tutorijale koji pomažu korisnicima da maksimalno iskoriste mogućnosti platforme.

Uz sve ove usluge, Hugging Face također nudi API-je koji omogućuju jednostavnu integraciju NLP rješenja u aplikacije. Ovi API-jevi omogućuju programerima da dodaju funkcionalnosti poput strojnog prevođenja, sažimanja teksta i analize sentimenta u svoje aplikacije bez potrebe za dubokim razumijevanjem NLP tehnologija. Ovaj pristup omogućuje brzu primjenu naprednih NLP rješenja u raznim industrijama.

4.1.2. OpenAI

OpenAI [2] je istraživačka organizacija koja se bavi razvojem naprednih jezičnih modela i tehnologija umjetne inteligencije. Osnovana 2015. godine, OpenAI je postala jedna od vodećih institucija u području umjetne inteligencije, poznata po svojim revolucionarnim jezičnim modelima kao što su GPT-3 i GPT-4.

Jedna od najvažnijih tehnologija koju OpenAI nudi je njihova serija generativnih predtreniranih transformera (engl. *generative pre-trained transformers, GPTs*) [13]. Ovi modeli koriste napredne metode dubokog učenja za generiranje prirodnog jezika na temelju velikih količina tekstualnih podataka. GPT-3, na primjer, koristi 175 milijardi parametara za generiranje tekstualnih odgovora, što ga čini jednim od najmoćnijih jezičnih modela dostupnih danas.

OpenAI API omogućuje programerima i istraživačima pristup moćnim AI modelima putem jednostavnog programskog sučelja. Njihov API podržava širok spektar NLP zadataka, uključujući generiranje teksta, odgovaranje na pitanja, strojno prevođenje, sažimanje teksta i mnoge druge. OpenAI stalno unapređuje svoj API kako bi korisnicima pružio više kontrole i fleksibilnosti u primjeni AI modela.

Jedna od ključnih značajki OpenAI API-ja je mogućnost finog podešavanja predtreniranih modela za specifične zadatke. Korisnici mogu prilagoditi modele kako bi bolje odgovarali njihovim specifičnim potrebama. Nove usluge uključuju kreiranje kontrolnih točaka (engl. *checkpoints*) tijekom svake epohe treniranja, što smanjuje potrebu za ponovnim treniranjem u slučajevima pretreniranja (engl. *overfitting*), te usporedni prikaz performansi različitih modela kako bi se olakšala evaluacija rezultata. [17] Ova poboljšanja omogućuju preciznije i učinkovitije treniranje modela.

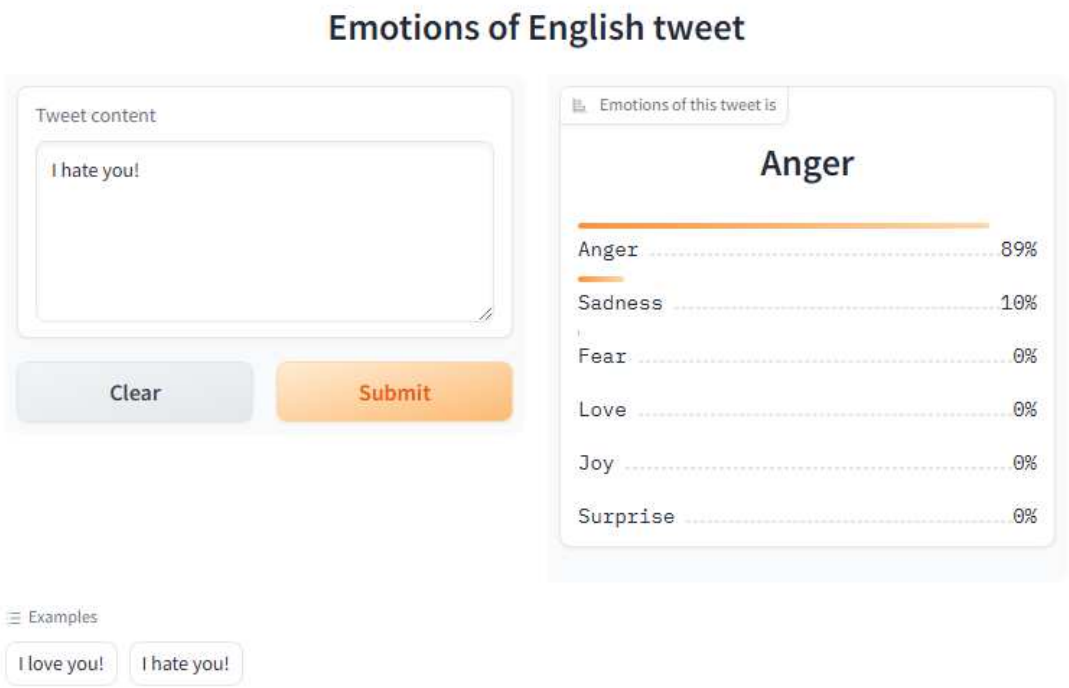
Administrativna kontrola je još jedno važno područje u kojem je OpenAI unaprijedio svoj API. Nova usluga 'Projects' omogućuje organizacijama preciznu kontrolu nad projektima, uključujući postavljanje uloga i API ključeva za specifične projekte,

ograničavanje pristupa određenim modelima, te postavljanje limita korištenja kako bi se izbjegli nepredviđeni troškovi. [16] Ove mogućnosti daju organizacijama veću fleksibilnost i sigurnost u upravljanju njihovim AI resursima.

Uz ove tehničke aspekte, OpenAI se također posvećuje etičkoj i sigurnoj primjeni umjetne inteligencije. Organizacija implementira mjere za sprječavanje zloupotrebe svojih tehnologija i osigurava da modeli generiraju odgovore koji su točni i prikladni za različite kontekste. OpenAI pruža opsežnu dokumentaciju i resurse za učenje kako bi korisnicima pomogli da maksimalno iskoriste njihove tehnologije. Ovi resursi uključuju tutorijale, vodiče za implementaciju, tehničke dokumente i primjere koda, što korisnicima omogućuje brzo i učinkovito uvođenje AI rješenja u njihove projekte.

4.1.3. Gradio

Gradio [5] je Python biblioteka koja omogućuje korisnicima da brzo i jednostavno demonstriraju svoje modele strojnog učenja putem web sučelja. Dizajnirana da bude pristupačna i jednostavna za korištenje, čak i za korisnike bez opsežnog tehničkog znanja, čime se omogućava širok raspon upotrebe u različitim industrijama i kontekstima.



Slika 4.1: Primjer Gradio aplikacije (analiza sentimenta). [3]

Korisnici mogu kreirati sučelja koja se mogu ugraditi u Python bilježnice ili prikazati kao zasebne web stranice, čime se olakšava dijeljenje i interakcija s modelima.

Integracija s Hugging Face omogućava trajno postavljanje Gradio aplikacija na njihovu platformu Hugging Face Spaces, koja pruža besplatne web hosting usluge. Također pruža javni URL za dijeljenje sa suradnicima, što dodatno pojednostavljuje proces demonstracije modela.

Jedna od naprednih značajki Gradio biblioteke je podrška za stvaranje prilagođenih web aplikacija koristeći klasu `Blocks`. Ova klasa omogućuje korisnicima kontrolu nad rasporedom komponenti na stranici, upravljanje složenim tokovima podataka i ažuriranje svojstava komponenti na temelju interakcije korisnika. Ova fleksibilnost čini Gradio izuzetno moćnim alatom za izradu složenih i prilagođenih aplikacija.

4.1.4. SelfCheckGPT biblioteka

Metoda SelfCheckGPT i sve njene varijante (BERTScore, QA, n-gram, NLI, LLM-Prompt) opisane su u prethodnom poglavlju. Istoimena Python biblioteka [14] nudi njihovu implementaciju. Sve varijante imaju metodu `predict` koja provjerava je li generirani odlomak haluciniran tako da vraća brojčanu vrijednost za svaku rečenicu. Visoka vrijednost ukazuje na halucinaciju. Biblioteka omogućava fleksibilnu integraciju s različitim jezičnim modelima, uključujući modele dostupne na Hugging Face platformi, kao što su Gemma i LLama, te OpenAI modele poput GPT-3.5 Turbo i GPT-4.

4.2. Korišteni modeli

4.2.1. Gemma 1.1 7B (IT)

Gemma modeli predstavljaju seriju laganih (engl. *lightweight*), naprednih jezičnih modela otvorenog koda (engl. *open-source*) koje je razvio Google na temelju tehnologije korištene za stvaranje Gemini modela. Ovi modeli dostupni su na engleskom jeziku u predtreniranim varijantama i varijantama prilagođenim za instrukcije. [4]

Gemma modeli su prikladni za različite zadatke generiranja teksta, uključujući odgovaranje na pitanja, sažimanje teksta i zaključivanje. Jedna od ključnih prednosti Gemma modela je njihova prilagodljivost i fleksibilnost. Modeli su dizajnirani tako da budu učinkoviti i u okruženjima s ograničenim resursima, omogućujući istraživačima i inženjerima da ih koriste bez potrebe za velikim računalnim resursima. Ovo čini Gemma modele pristupačnijim široj zajednici, potičući inovacije i primjenu naprednih AI tehnologija u raznim industrijama.

Gemma 1.1 7B (IT) [4] je verzija Gemma modela posebno prilagođena za instrukcije. Predstavlja ažuriranje originalnog Gemma modela prilagođenog za instrukcije, s nizom poboljšanja i optimizacija. Gemma 1.1 treniran je pomoću nove metode učenja, učenje podrškom na osnovu povratnih informacija od ljudi (engl. *reinforcement learning from human feedback, RLHF*), što je dovelo do značajnih poboljšanja u kvaliteti, sposobnostima kodiranja, činjeničnoj točnosti i praćenju uputa. Ova metoda omogućuje modelu da bolje razumije i odgovara na složene upite, generirajući relevantne i točne odgovore.

4.2.2. Meta Llama 3 8B Instruct

Model Meta Llama 3 8B Instruct [10] dio je obitelji velikih jezičnih modela koje je razvila Meta. Koristi samo tekst kao ulaz te generira tekst i kod kao izlaz. Meta nudi ovu kolekciju predtreniranih modela i modela podešenih za instrukcije u dvije veličine (8 i 70 milijardi parametara). Ovi modeli namijenjeni su za komercijalnu i istraživačku upotrebu na engleskom jeziku. Modeli prilagođeni za instrukcije namijenjeni su za chatbot asistente, dok se predtrenirani modeli mogu prilagoditi za različite zadatke generiranja prirodnog jezika.

Llama 3 modeli su autoregresivni (engl. *auto-regressive*) jezični modeli koji koriste optimiziranu transformer arhitekturu. Posebno su korisni za dijalog zbog njihove sposobnosti generiranja koherentnih i korisnih odgovora na upite korisnika. Korištenje RLHF metode i nadziranog finog podešavanja (engl. *supervised fine-tuning, SFT*) omogućuje modelima da bolje razumiju i slijede upute, što ih čini idealnim za primjene koje zahtijevaju visoku razinu interakcije i prilagodljivosti. Osim toga, predtrenirani modeli mogu se koristiti za razne zadatke, uključujući generiranje koda, što proširuje njihovu primjenjivost u različitim domenama.

Tijekom razvoja ovih modela, Meta je posvetila veliku pažnju optimizaciji korisnosti i sigurnosti. Optimizacija modela za sigurnost i korisnost osigurava da generirani odgovori budu ne samo točni, već i prikladni za različite kontekste upotrebe. Ovo je posebno važno u komercijalnim aplikacijama gdje je interakcija s korisnicima ključna za pružanje kvalitetne usluge.

4.2.3. Mistral 7B Instruct v0.3

U suradnji s Hugging Face-om, Mistral AI je predstavio Mistral 7B Instruct v0.3 [11], napredni jezični model koji koristi transformer arhitekturu i posebno je prilagođen za

zadatke generiranja teksta temeljenih na uputama. Ovaj model predstavlja sljedeću iteraciju Mistral 7B modela, s nizom ključnih poboljšanja.

Jedna od najvažnijih nadogradnji u ovoj verziji je proširenje vokabulara na 32 768 tokena, što omogućava modelu da razumije i generira širi spektar riječi i fraza. Ovo proširenje vokabulara značajno poboljšava sposobnost modela za obradu jezika, omogućujući mu bolje razumijevanje konteksta i generiranje preciznijih odgovora.

Mistral 7B Instruct v0.3 također uvodi funkcionalnost pozivanja funkcija tijekom obrade jezika. Ovo modelu omogućava izvršavanje unaprijed definiranih funkcija tijekom generiranja odgovora, što je posebno korisno u scenarijima dinamičke interakcije i manipulacije podacima u stvarnom vremenu.

Evaluacije performansi Mistral 7B Instruct v0.3 modela pokazale su značajna poboljšanja u odnosu na prethodne verzije. Model je pokazao izvanrednu sposobnost generiranja koherentnog i kontekstualno prikladnog teksta temeljenog na korisničkim uputama. Unatoč ovim poboljšanjima, važno je napomenuti da trenutna verzija modela nema mehanizme za moderiranje sadržaja. To se treba uzeti u obzir pri implementaciji u okruženjima koja zahtijevaju moderirane izlaze kako bi se izbjegao neprikladan ili štetan sadržaj. [33]

4.2.4. Nous Hermes 2 Mixtral 8x7B DPO

Nous Research razvio je model Nous Hermes 2 Mixtral 8x7B DPO [12], napredni jezični model s ciljem pružanja visokih performansi u raznim zadacima obrade prirodnog jezika. Ovaj model koristi kombinaciju nadziranog finog podešavanja i direktne optimizacije preferenci (engl. *direct preference optimization, DPO*), što mu omogućuje da postigne vrhunske rezultate u različitim okruženjima.

Model je treniran na više od milijun podataka, uglavnom generiranih pomoću GPT-4, zajedno s drugim visokokvalitetnim podacima iz različitih skupova podataka u području umjetne inteligencije. Ovaj opsežan i raznovrstan skup podataka omogućio je modelu da postigne vrhunske performanse u zadacima poput odgovaranja na pitanja, razumijevanja teksta i generiranja koda.

Nous Hermes 2 Mixtral 8x7B DPO koristi Mixtral 8x7B arhitekturu s tehnikom mješavine stručnjaka (engl. *mixture of experts, MoE*), što doprinosi njegovoj sposobnosti da se prilagodi različitim kontekstima te generira precizne i koherentne odgovore. Jedna od ključnih značajki ovog modela je uvođenje ChatML (Chat Markup Language) formata upita, koji omogućava bolju strukturu i interakciju tijekom razgovora s modelom. ChatML format pruža mogućnost definiranja pravila, uloga i stilova, što po-

boljšava korisničko iskustvo i omogućava preciznije usmjeravanje odgovora modela. [36]

4.2.5. GPT-3.5 Turbo

GPT-3.5 Turbo [13] je napredna verzija GPT-3 modela, dizajnirana da bude brža i efikasnija. Ovaj model koristi kontekstni prozor (engl. *context window*) od 16 000 tokena, što omogućuje obradu većih količina teksta u jednom zahtjevu. Ova poboljšanja čine ga idealnim za aplikacije koje zahtijevaju obradu velikih tekstualnih podataka. Osim toga, GPT-3.5 Turbo podržava pozivanje funkcija, što omogućuje modelu da izvršava definirane funkcije i uključi rezultate u svoje odgovore. Ova funkcionalnost je korisna za zadatke kao što su generiranje koda ili pristup vanjskim API-jima. Fine podešavanje ovog modela omogućava korisnicima prilagodbu modela za specifične potrebe, poboljšavajući performanse i pouzdanost formata odgovora.

4.2.6. GPT-4

GPT-4 [13] predstavlja značajan napredak u odnosu na prethodne verzije, pružajući poboljšane mogućnosti razumijevanja i generiranja prirodnog jezika. Model je sposoban za složenije zadatke generiranja teksta i ima bolju sposobnost praćenja uputa u odnosu na svoje prethodnike. GPT-4 je također optimiziran za sigurnost i etičku upotrebu, smanjujući rizik od generiranja štetnih ili neprimjerenih sadržaja. Osim toga, GPT-4 koristi veći kontekstni prozor od 32 000 tokena, omogućujući obradu duljih tekstualnih ulaza u jednom zahtjevu.

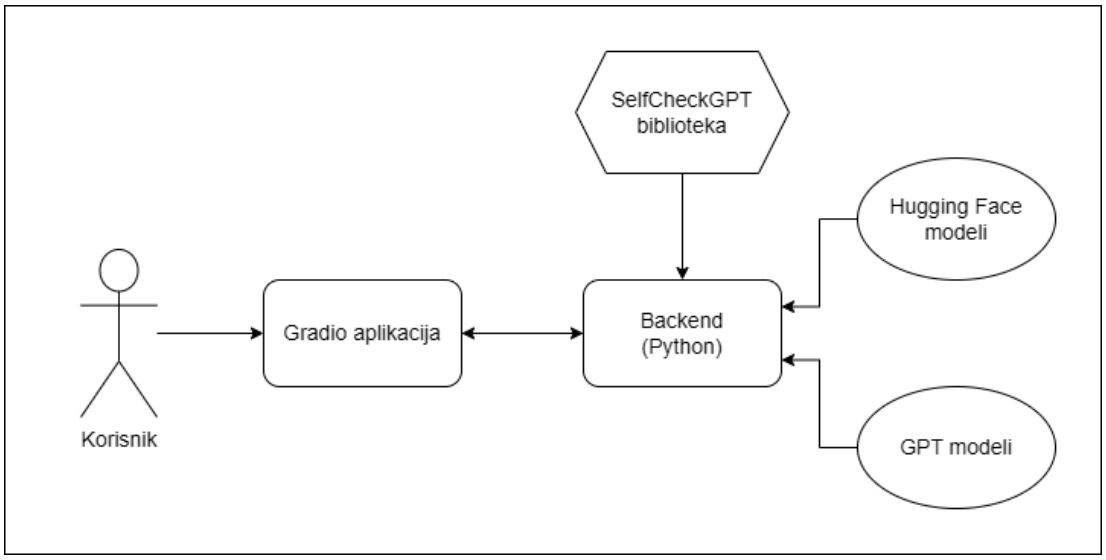
4.2.7. GPT-4 Turbo

GPT-4 Turbo [13] je unaprijeđena verzija GPT-4 modela, dizajnirana da bude brža i ekonomičnija. Ovaj model koristi kontekstni prozor od 128 000 tokena, što je značajno povećanje u odnosu na standardni GPT-4. Ovo omogućuje modelu da obradi još veće količine teksta u jednom zahtjevu, čineći ga idealnim za aplikacije koje zahtijevaju obradu velikih skupova podataka. GPT-4 Turbo također podržava pozivanje funkcija i integraciju s vanjskim alatima, omogućujući složenije interakcije i bolje performanse u specifičnim zadacima.

4.2.8. GPT-4o

GPT-4o [13] je najnovija verzija modela u GPT seriji, dizajniran da bude najbrži i najefikasniji model u seriji. Ovaj model kombinira visoku performanse s niskim troškovima, čineći ga idealnim za široku primjenu u različitim industrijama. GPT-4o je optimiziran za brze odgovore i može se koristiti za razne zadatke, od generiranja teksta do složenih analiza podataka. Ova verzija također podržava sve napredne značajke prethodnih modela, uključujući pozivanje funkcija i prošireni kontekstni prozor.

4.3. Opis rada sustava



Slika 4.2: Skica rada sustava.

Sustav je razvijen u programskom jeziku Python, temelji se na korištenju naprednih jezičnih modela i tehnologija za detekciju halucinacija u generiranom tekstu. Aplikacija korisnicima omogućuje korištenje chatbota i detekciju halucinacija u odgovorima chatbota. Korisnici mogu birati između ponuđenih modela i promijeniti postavke za generiranje teksta i detekciju halucinacija putem interaktivnog korisničkog sučelja koje je kreirano koristeći Gradio. Primjer korištenja i specifične komponente aplikacije prikazane su i opisane u sljedećem poglavlju.

4.4. Prilagodba SelfCheckGPT metoda

Tijekom razvoja aplikacije prilagođene su određene metode iz SelfCheckGPT biblioteke. Sve SelfCheckGPT varijante, osim one s n-gramom, vraćaju ocjenu halucinacije u intervalu $[0, 1]$. Vrijednost koju vraća SelfCheckGPT s n-gramom je neograničena, tj. nalazi se u intervalu $[0, +\infty)$. Preslikana je u odgovarajući interval pomoću monotone funkcije.

Kod 4.1: LLM Prompt metoda [14]

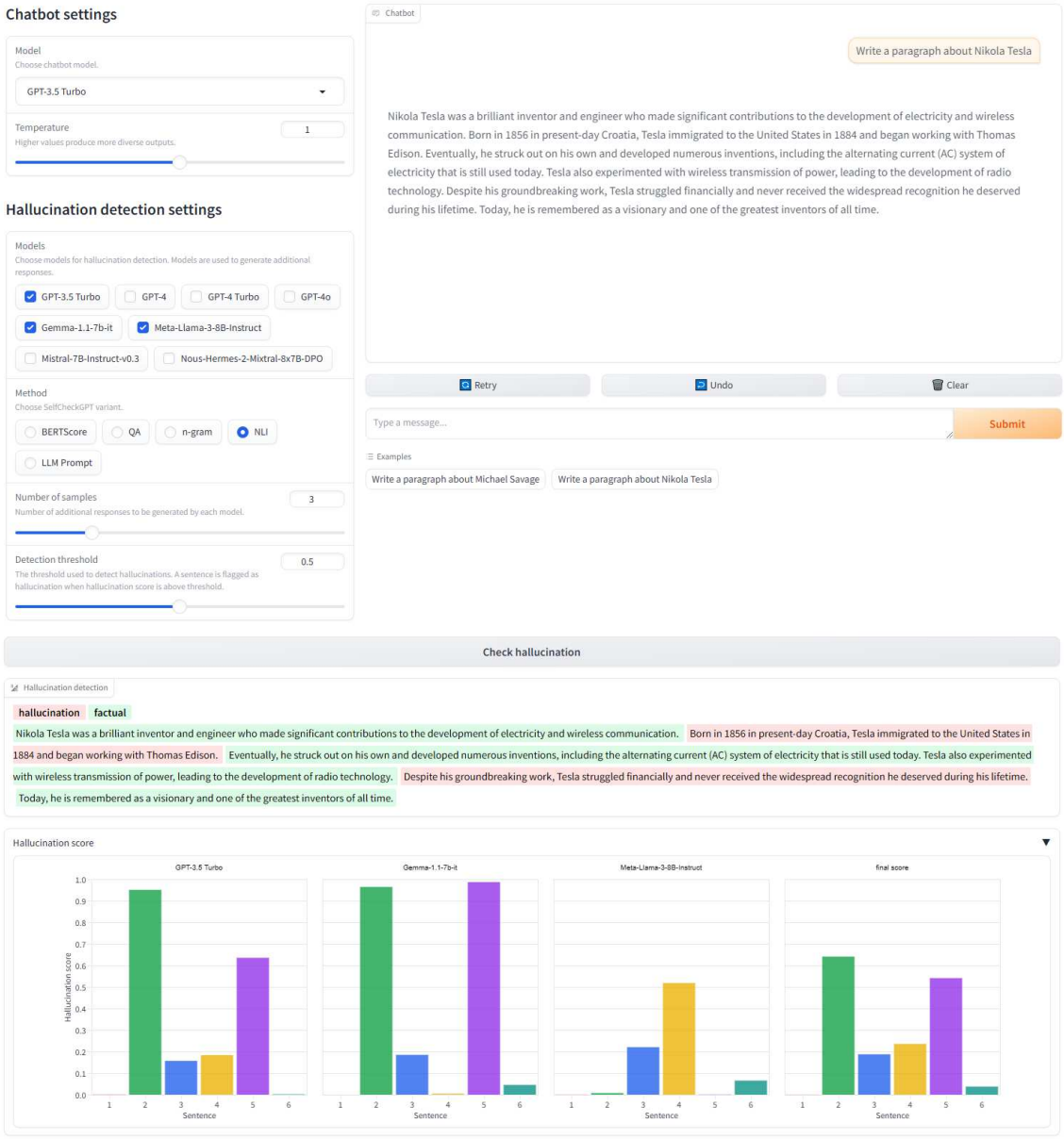
```
class SelfCheckLLMPrompt:
    def __init__(...):
        ...
        self.text_mapping = {'yes': 0.0, 'no': 1.0, 'n/a': 0.5}
        self.not_defined_text = set()

    def text_postprocessing(self, text,):
        """
        To map from generated text to score
        Yes -> 0.0
        No  -> 1.0
        everything else -> 0.5
        """

        text = text.lower().strip()
        if text[:3] == 'yes':
            text = 'yes'
        elif text[:2] == 'no':
            text = 'no'
        else:
            if text not in self.not_defined_text:
                print(f"warning: {text} not defined")
                self.not_defined_text.add(text)
            text = 'n/a'
        return self.text_mapping[text]
```

LLM Prompt varijanta je također trebala biti prilagođena. Na slici 3.5 vidimo kako funkcionira LLM Prompt metoda. Odabranom LLM-u direktno postavljamo upit "Podupire li uzorak rečenicu?" za svaki uzorak i svaku rečenicu, te očekujemo da će odgovoriti s "Da" ili "Ne". Isječak koda 4.1 prikazuje relevantan kod iz SelfCheckGPT

biblioteke. Vidimo da se odgovoru "yes" pridjeljuje vrijednost 0.0, odgovoru "no" vrijednost 1.0, a svi ostali mogući odgovori se označavaju kao "n/a" i boduju s 0.5. Uočite da se "yes" ili "no" mora nalaziti na početku odgovora LLM-a.

Tijekom razvoja aplikacije, uočeno je da se neki odgovori svrstavaju u "n/a" kategoriju, iako sadrže riječ "yes" ili "no". Problem je bio u stilizaciji teksta (engl. *text stylization*). Naime, neki LLM-ovi žele istaknuti tekst tako da ga podebljaju ili na neki drugi način urede. Zbog toga, odgovor LLM-a može izgledati ovako: `<b>Yes</b>` ili `**Yes**`. Riječ "yes" se ne nalazi direktno na početku odgovora, stoga se on svrstava kao "n/a". Kako bi popravili ovo pogrešno svrstavanje, originalan kod je izmjenjen tako da se riječ "yes" ili "no" može nalaziti bilo gdje u tekstu, ne samo na početku. Generirani tekst će uvijek biti kratak jer je odgovor LLM-a ograničen na maksimalno 5 tokena.

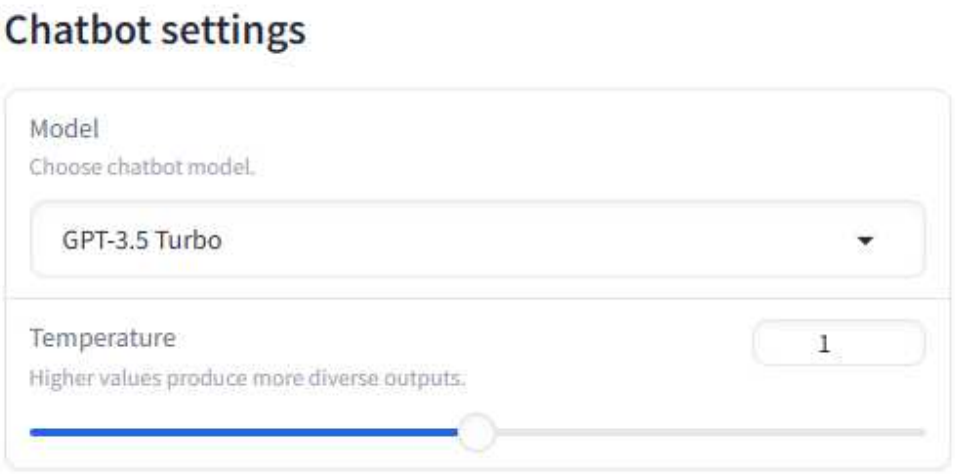
5. Aplikacija



Slika 5.1: Izgled aplikacije.

5.1. Komponente

5.1.1. Postavke za chatbot



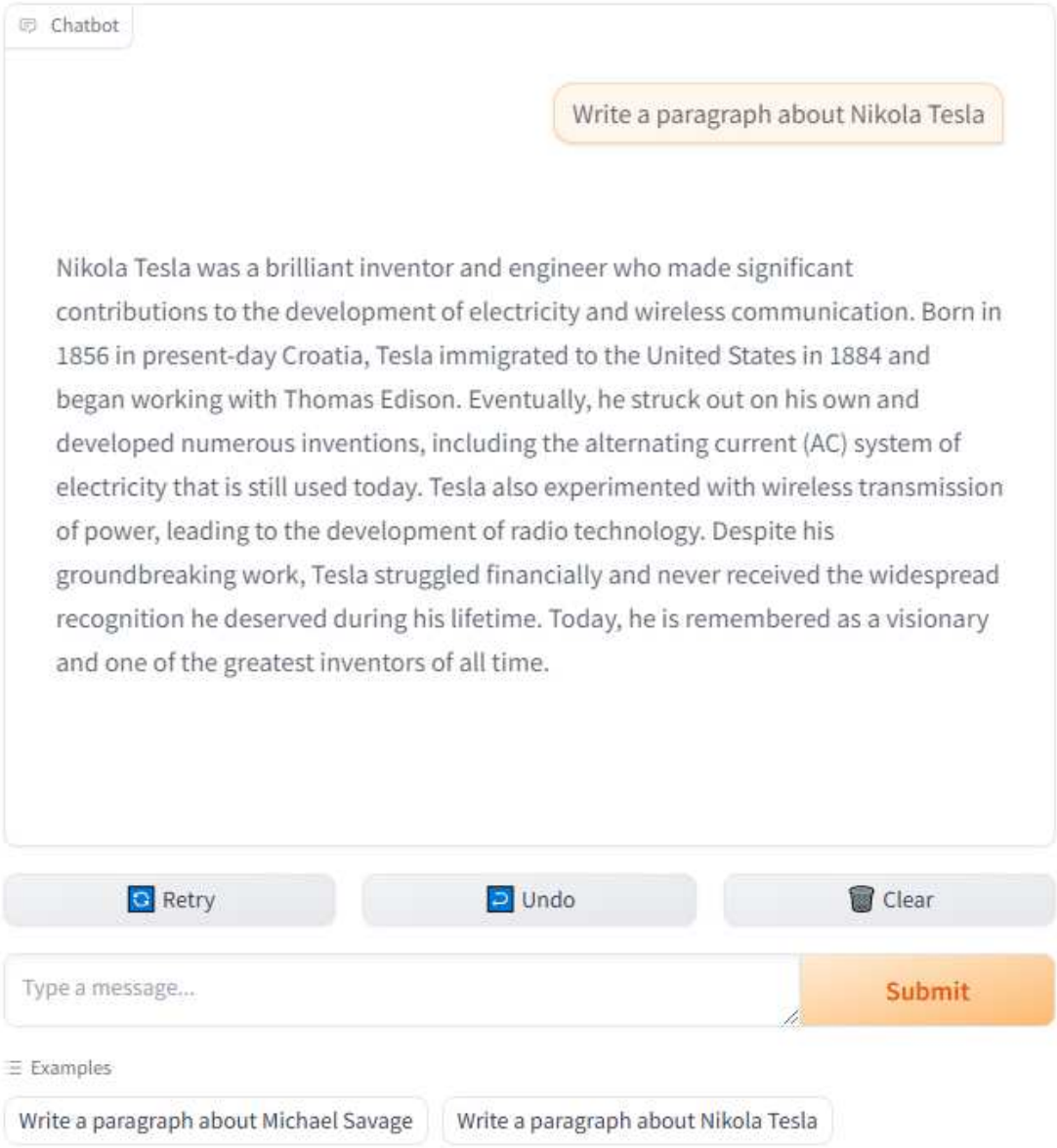
Slika 5.2: Postavke za chatbot.

Ova sekcija aplikacije omogućuje korisnicima prilagodbu postavki za generiranje odgovora chatbota. Korisnik odabire specifičan jezični model iz padajućeg izbornika (engl. *dropdown menu*) koji će chatbot koristiti za odgovaranje na upite korisnika. Dostupni modeli su navedeni i opisani u prošlom poglavlju.

Korisnik također može podesiti temperaturu pomoću klizača (engl. *slider*). On omogućuje korisnicima kontrolu nad raznolikošću generiranih odgovora chatbota. Vrijednosti temperature se nalaze u intervalu $[0, 2]$, pri čemu niže vrijednosti temperature (bliže 0) rezultiraju deterministčkim i konzistentnim odgovorima, jer model bira najvjerojatnije riječi za generiranje teksta. S druge strane, više vrijednosti temperature (bliže 2) povećavaju raznolikost odgovora, čineći tekst kreativnijim i više nepredvidljivim, jer model eksperimentira s manje vjerojatnim riječima.

5.1.2. Chatbot

Chatbot komponenta je izgrađena pomoću Gradio sučelja `ChatInterface`. Ova komponenta omogućuje korisnicima da lako komuniciraju s odabranim modelom u stvarnom vremenu. Chatbot radi u streaming načinu rada, što znači da se odgovor prikazuje token po token, kako ih model generira. To omogućuje prikazivanje postepene generiranje odgovora, umjesto čekanja da se cijeli odgovor generira.

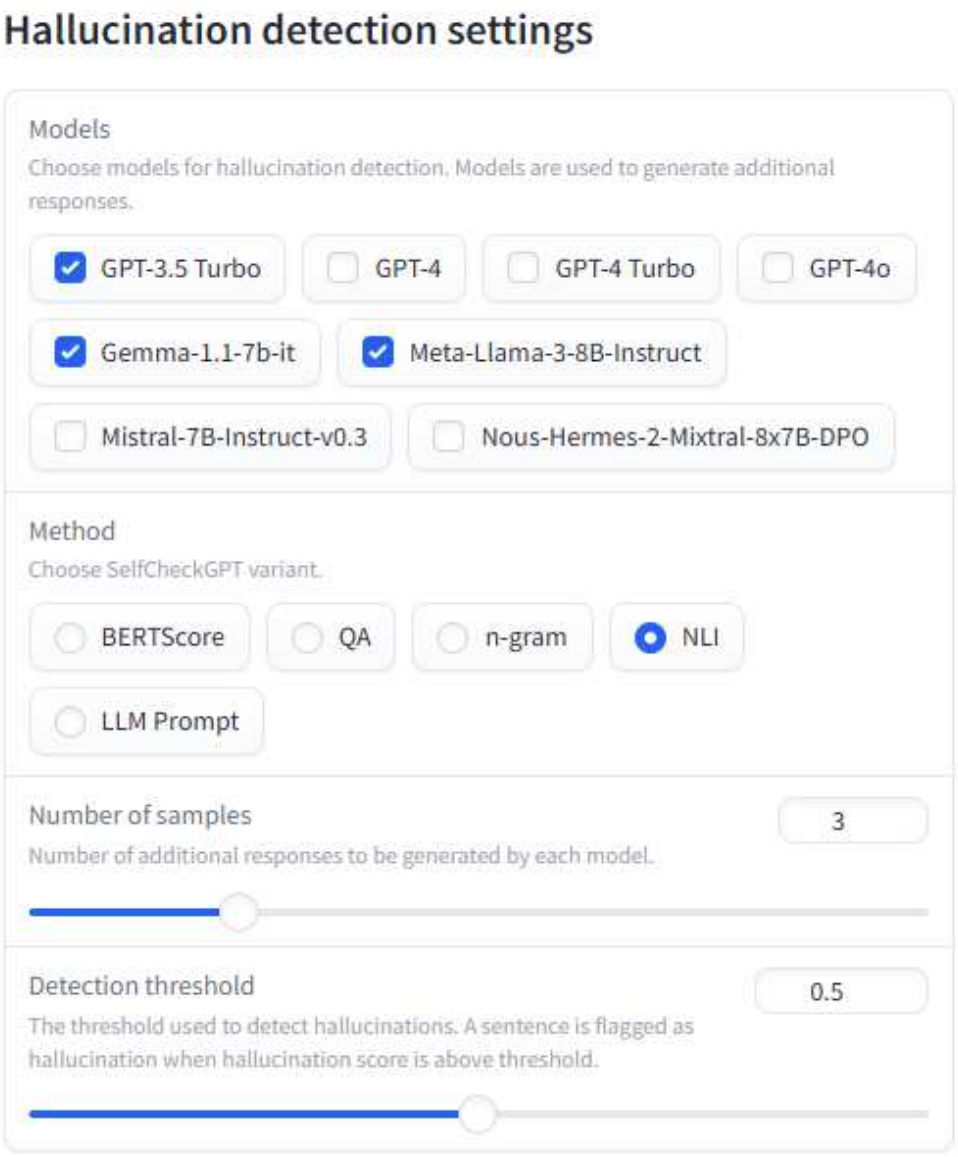


Slika 5.3: Chatbot.

Korisnik postavlja upit tako da ga upiše u tekstno polje i klikne gumb "Submit" ili stisne tipku Enter. Također može odabrati jedan od ponuđenih primjera upita. Tijekom generiranja odgovora, gumb "Submit" je zamijenjen gumbom "Stop". Klikom na taj gumb prekida se generiranje odgovora. Na slici 5.3 vidljiv je upit "Napiši odlomak o Nikoli Tesli" i odgovor chatbota.

Klikom na gumb "Retry" pokreće se ponovno generiranje odgovora na zadnji korisnikov upit. Gumb "Undo" briše zadnji korisnikov upit i odgovor chatbota, a gumb "Clear" briše cijeli razgovor.

5.1.3. Postavke za detekciju halucinacija



Slika 5.4: Postavke za detekciju halucinacija.

Ova sekcija aplikacije omogućuje korisnicima prilagodbu postavki za detekciju halucinacija. Korisnik odabire jedan ili više jezičnih modela koji će se koristiti za generiranje dodatnih odgovora tijekom procesa detekcije halucinacija. Ovo je korisno za usporedbu performansi različitih modela i kombiniranje njihovih rezultata radi postizanja točnijih procjena. Popis dostupnih modela vidljiv je na slici 5.4. Isti modeli su navedeni i u padajućem izborniku u postavkama za chatbot.

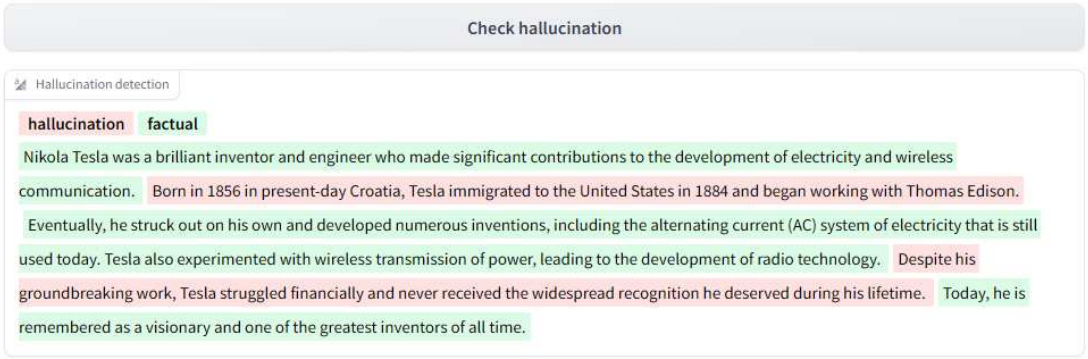
Korisnik zatim odabire jednu od SelfCheckGPT varijanti, tj. odabire specifičnu metodu za detekciju halucinacija. Varijante SelfCheckGPT-a (BERTScore, QA, n-

gram, NLI, LLMPrompt) opisane su u prethodnom poglavlju. Svaka metoda nudi jedinstveni pristup evaluaciji, omogućujući korisnicima fleksibilnost u odabiru najprikladnijeg alata za svoje specifične potrebe.

Korisnik može odabrati broj dodatnih odgovora, tj. uzoraka (engl. *samples*) koje će odabrani modeli generirati za proces samoprovjere (engl. *self-check*). Na primjer, postavljanjem klizača na vrijednost 3, svaki odabrani model će generirati po 3 uzorka. Veći broj uzoraka može poboljšati točnost procjene, ali također produžuje vrijeme obrade.

Sljedeći klizač postavlja prag za detekciju (engl. *detection threshold*) halucinacija. Ocjena halucinacije (engl. *hallucination score*) nalazi se u intervalu $[0, 1]$. Ako je ocjena halucinacije iznad postavljenog praga, rečenica će biti označena kao halucinacija. Na primjer, ako je prag postavljen na 0.5, sve rečenice s ocjenom halucinacije iznad te vrijednosti bit će označene kao halucinacije. Ova funkcionalnost omogućuje prilagodbu osjetljivosti sustava za detekciju halucinacija, što je korisno za različite kontekste i zahtjeve.

5.1.4. Prikaz rezultata

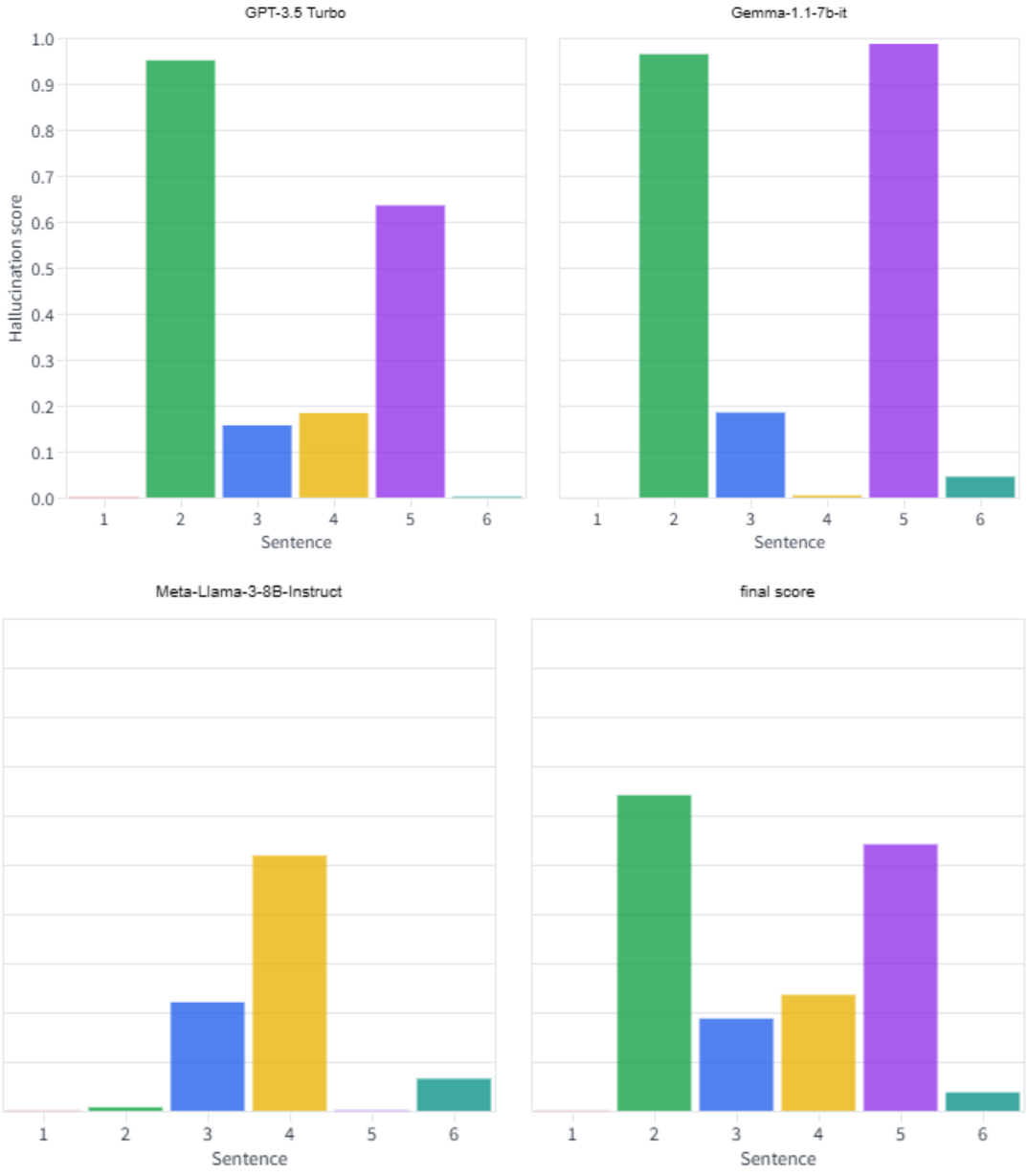


Slika 5.5: Gumb za pokretanje detekcije halucinacija i prikaz označenog teksta.

Nakon što je korisnik odabrao postavke za detekciju halucinacija, može pokrenuti proces detekcije halucinacija pomoću gumba "Check hallucination". Kad je gotov izračun ocjena halucinacije, rezultat se prikazuje kao označeni tekst. Kao što je vidljivo na slici 5.5, halucinacije su označene crvenom bojom, a točne informacije zelenom bojom. Halucinacije se detektiraju na razini rečenice.

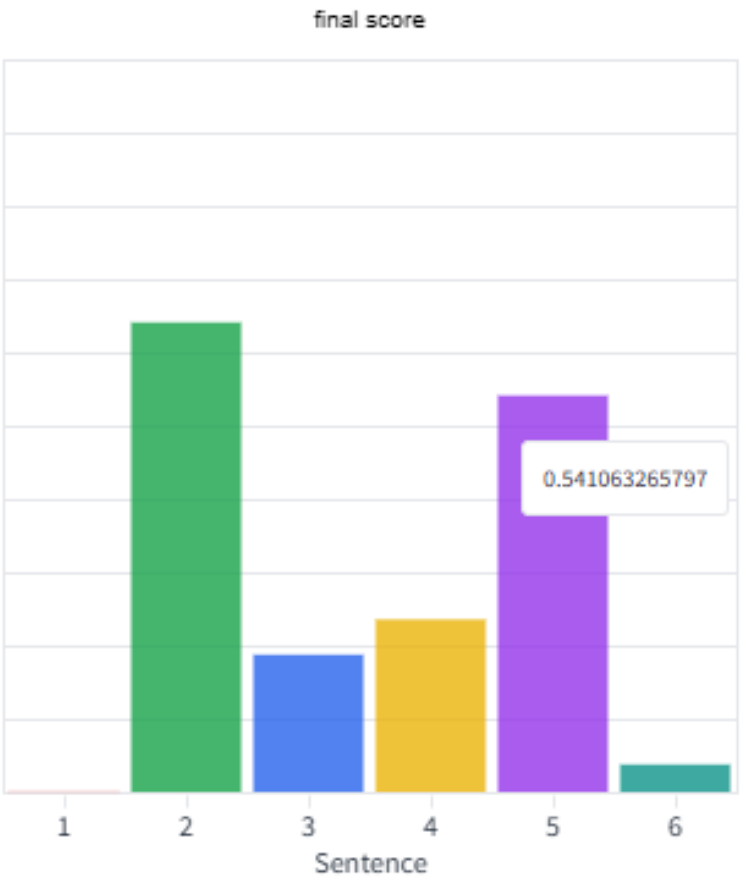
Prikaz rezultata uključuje grafički prikaz pomoću stupčastog dijagrama (engl. *bar plot*) za svaki odabrani model i za konačne ocjene halucinacije. Naime, svaki model generira uzorke koji doprinose konačnoj ocjeni halucinacije. Ovi dijagrami služe za vizualizaciju doprinosa svakog odabranog modela. Svaki dijagram prikazuje ocjenu

halucinacije za svaku rečenicu iz odlomka kojeg je generirao chatbot. Zadnji dijagram prikazuje konačnu ocjenu halucinacije za svaku rečenicu. Konačna ocjena je izračunata kao prosjek ocjena svih odabranih modela. X-os prikazuje redni broj rečenice, a y-os prikazuje ocjenu halucinacije.

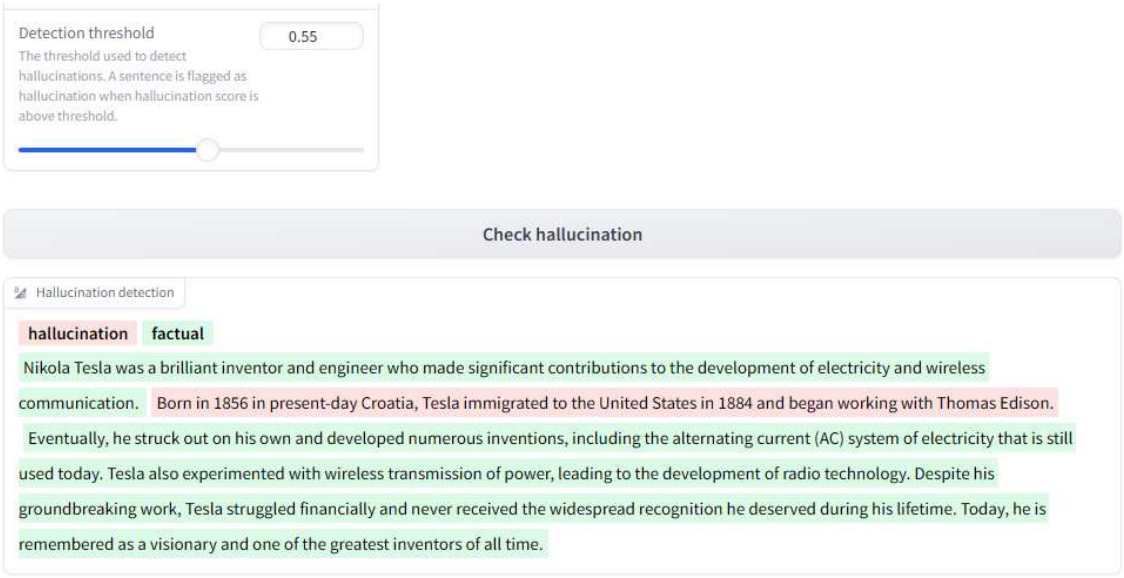


Slika 5.6: Stupčasti dijagrami.

Na slici 5.7 vidimo da peta rečenica ima konačnu ocjenu halucinacije od oko 0.54. Prag detekcije je postavljen na 0.5, stoga je ta rečenica označena kao halucinacija na slici 5.5. Ako pomaknemo prag na 0.55, peta rečenica će biti označena kao točna (vidljivo na slici 5.8).

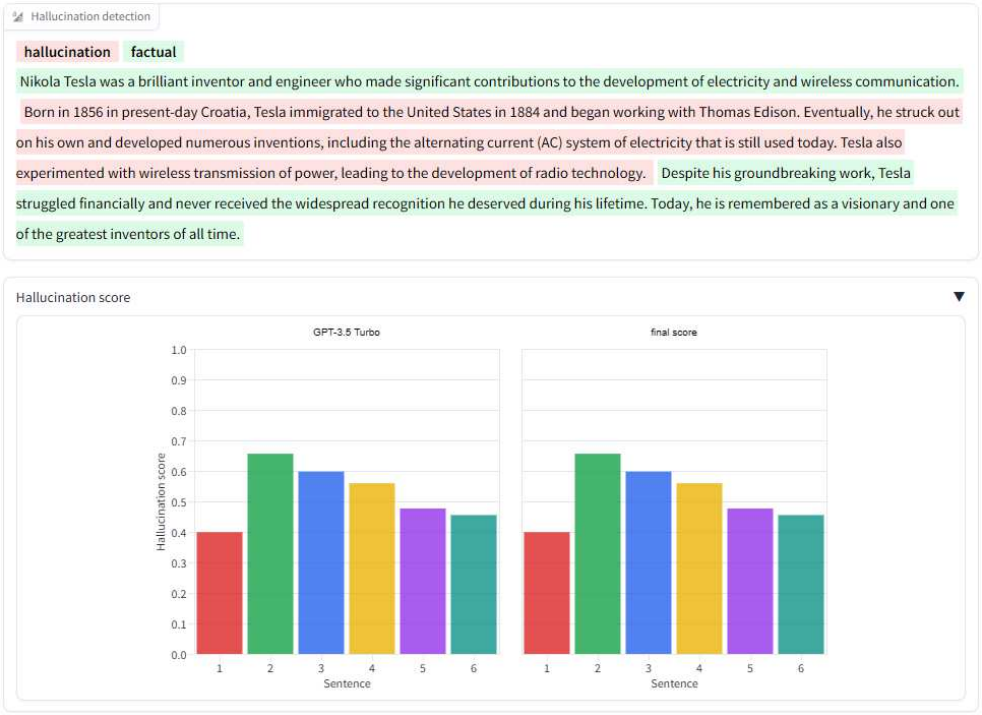


Slika 5.7: Stupčasti dijagram za konačne ocjene halucinacije.

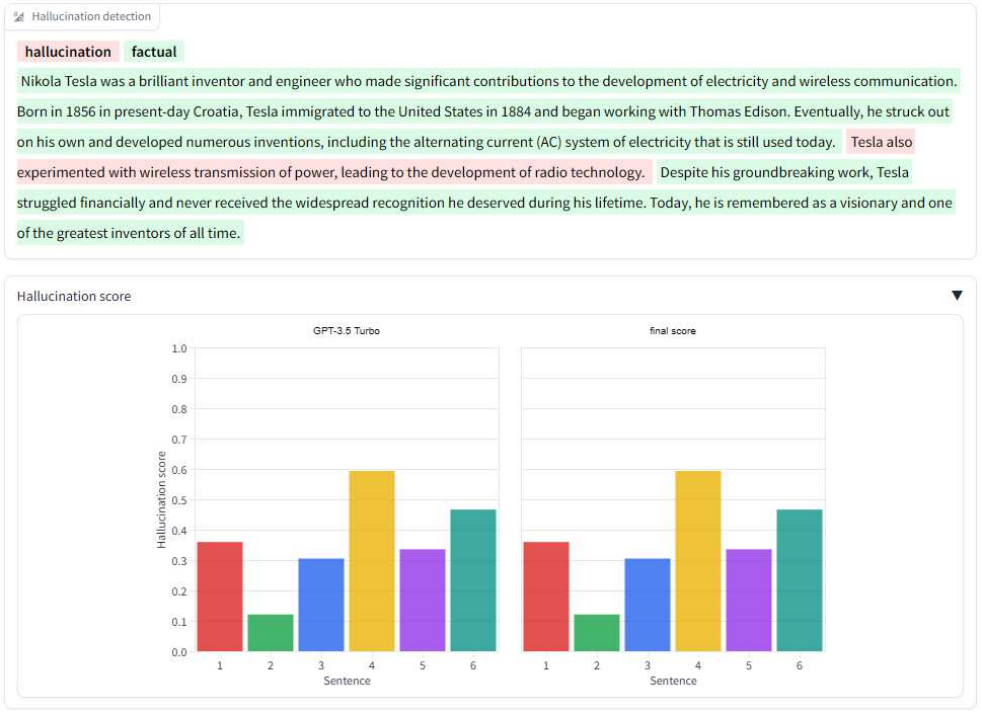


Slika 5.8: Prikaz označenog teksta nakon promjene praga detekcije.

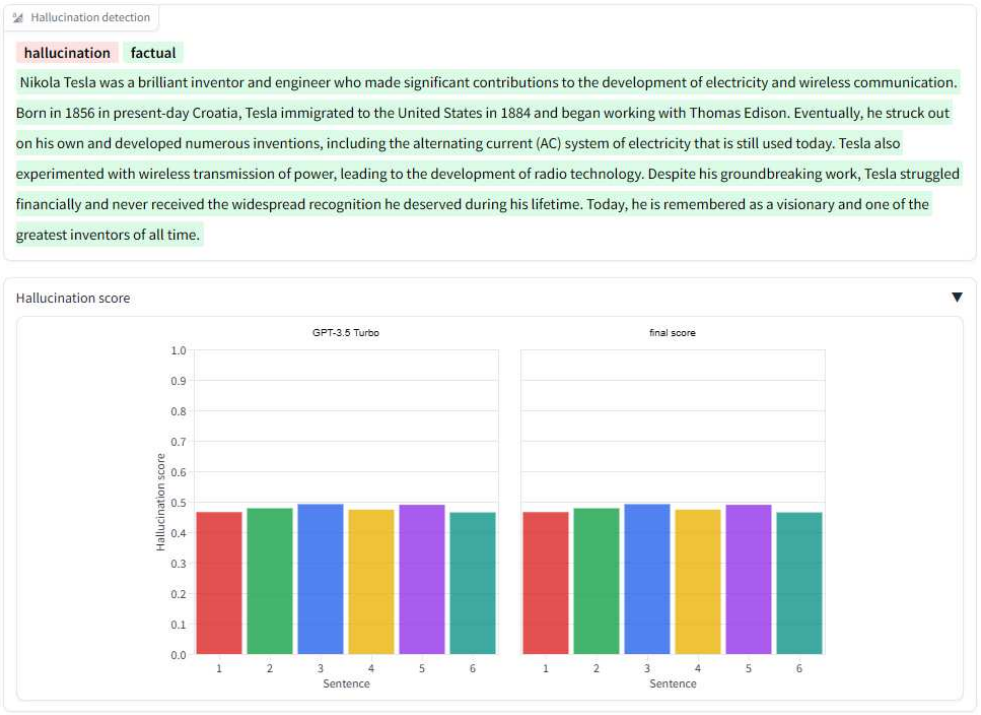
5.2. Usporedba rezultata različitih metoda



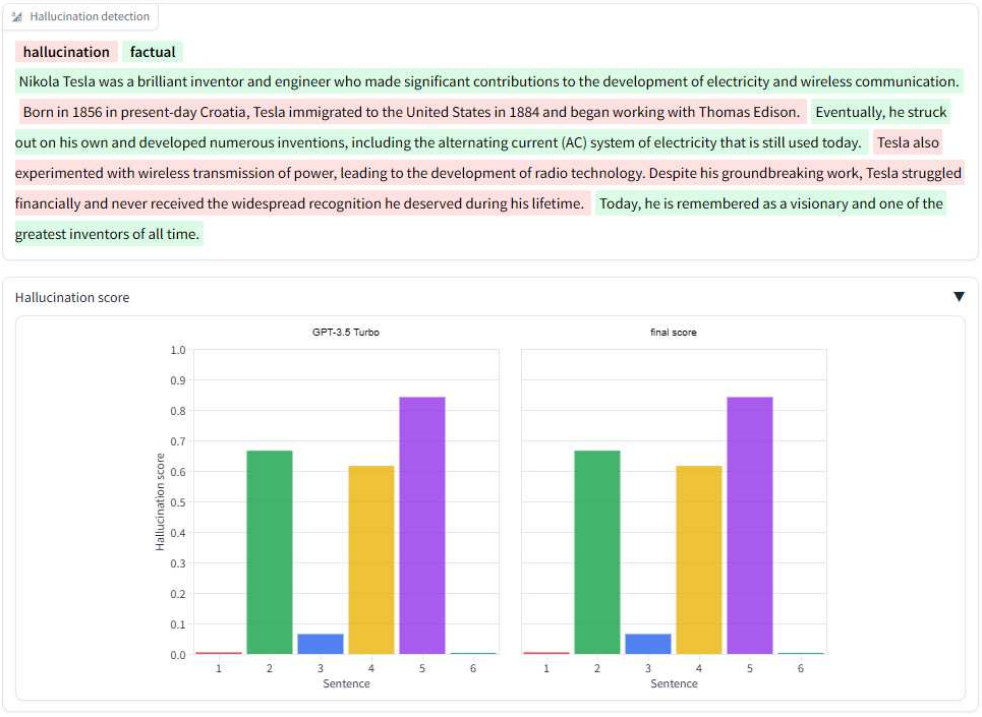
Slika 5.9: Detekcija halucinacija s BERTScore varijantom.



Slika 5.10: Detekcija halucinacija s QA varijantom.



Slika 5.11: Detekcija halucinacija s n-gram varijantom.



Slika 5.12: Detekcija halucinacija s NLI varijantom.

Slika 5.13: Detekcija halucinacija s LLM Prompt varijantom.

6. Diskusija

Rezultati prijašnjih istraživanja su pokazali da SelfCheckGPT može učinkovito detektirati halucinacije u izlazima LLM-ova bez potrebe za vanjskim bazama podataka ili pristupom internim vjerojatnostima modela. Dok su Manakul i suradnici [29] evaluirali tekstove isključivo generirane GPT-3 modelom, naša web aplikacija nudi više različitih LLM-ova za generiranje odgovora chatbota. Također, korisnik može odabrati jedan model (koji može biti različit od chatbot modela) ili više modela za generiranje uzoraka. Kombiniranjem više izvora informacija nastojimo postići bolju procjenu i time unaprijediti detekciju halucinacija.

6.1. Ograničenja

Kvaliteta detekcije halucinacija uvelike ovisi o raznovrsnosti generiranih uzoraka. Kada modeli generiraju različite verzije odgovora, sustav može učinkovitije prepoznati neusklađenosti i identificirati potencijalne halucinacije. Ako model generira uzorke koji su vrlo slični ili identični originalnom odgovoru, sustav može imati problema s detekcijom halucinacija. U takvim slučajevima, sustav možda neće moći razlikovati stvarne informacije od halucinacija jer nema dovoljno varijacije u podacima za analizu. Raznolikost uzoraka se nastojala postići ponudom raznih LLM-ova i postavljanjem temperature na vrijednost 1.0 pri generiranju uzoraka.

Generiranje i analiza dodatnih uzoraka za procjenu halucinacija može zahtijevati značajno vrijeme obrade i računalne resurse, posebno kada se koriste složeni modeli i metode detekcije halucinacija. Ovo može biti značajno ograničenje u aplikacijama koje zahtijevaju brzu obradu podataka ili u okruženjima s ograničenim resursima.

Razvijen sustav detektira halucinacije na razini rečenice. Ovaj pristup je učinkovit u identificiranju potpuno netočnih rečenica, no neke rečenice mogu istovremeno sadržavati točne i netočne informacije. To predstavlja izazov za ovakav sustav jer ne može precizno odrediti koji dijelovi rečenice sadrže netočne informacije. Kada se detekcija vrši na razini cijele rečenice, postoji rizik da sustav ne može identificirati halucina-

cije koje su suptilno umetnute unutar inače točnog konteksta. Na primjer, rečenica može uglavnom sadržavati točne činjenice i podatke, ali također imati jedan ili više dijelova koji su izmišljeni ili pogrešno predstavljeni. Ovaj problem može dovesti do nepotpunog ili netočnog prepoznavanja halucinacija, smanjujući ukupnu učinkovitost sustava. Potrebno je razviti pristupe koji omogućuju finiju analizu unutar rečenica. To može uključivati metode koje omogućuju analizu po frazama ili čak pojedinačnim riječima. Na taj način, sustavi bi mogli identificirati specifične dijelove rečenica koji sadrže halucinacije, omogućujući precizniju detekciju i bolju interpretaciju rezultata.

7. Zaključak

Umjetna inteligencija i obrada prirodnog jezika imaju potencijal da značajno unaprijede mnoge aspekte našeg svakodnevnog života. Međutim, kako bi se ovaj potencijal u potpunosti iskoristio, potrebno je riješiti izazove poput halucinacija u velikim jezičnim modelima. Halucinacije predstavljaju značajan izazov jer mogu dovesti do generiranja netočnih ili zavaravajućih informacija, što može imati ozbiljne posljedice u domenama i primjenama gdje je točnost informacija ključna.

U ovom radu istražili smo problem halucinacija u velikim jezičnim modelima i razvili sustav za njihovu detekciju i upravljanje, koristeći SelfCheckGPT biblioteku. Razvili smo chatbot aplikaciju s integriranim sustavom za detekciju halucinacija, omogućujući korisnicima interakciju s različitim LLM-ovima. Aplikacija je osmišljena tako da korisnicima pruži intuitivno sučelje za interakciju s chatbotom koji koristi razne LLM-ove. Korisnik započinje sesiju biranjem jednog od dostupnih jezičnih modela za generiranje odgovora chatbota. Korisnici također mogu odabrati jedan ili više modela za generiranje uzoraka, što omogućuje bolju raznolikost uzoraka. Aplikacija nudi vizualni prikaz rezultata detekcije, ističući dijelove teksta koji su identificirani kao potencijalne halucinacije, te grafički prikaz doprinosa odabranih LLM-ova. Korisnici mogu pregledati ove rezultate i dobiti dodatne informacije o tome zašto je određeni dio teksta označen kao halucinacija. Ovaj vizualni prikaz pomaže korisnicima da bolje razumiju proces detekcije halucinacija i donose informirane odluke temeljene na rezultatima detekcije. Sustav također omogućuje prilagodbu osjetljivosti detekcije halucinacija, što je korisno za različite kontekste i zahtjeve korisnika.

Što se tiče ograničenja, potrebno je poboljšati detekciju suptilnih halucinacija koje mogu biti prisutne unutar inače točnih rečenica. Također, istraživanje metoda za finiju analizu unutar rečenica može pomoći u identifikaciji specifičnih dijelova teksta koji sadrže halucinacije. Kroz kontinuirano istraživanje i unapređenje metoda za detekciju halucinacija, možemo osigurati da veliki jezični modeli pružaju točne i pouzdane informacije. To će povećati povjerenje korisnika u sustave umjetne inteligencije i omogućiti širu primjenu ovih tehnologija u različitim domenama.

Kroz kontinuirano istraživanje i poštovanje etičkih smjernica, nastoji se postići ravnoteža između inovacije i odgovornosti, omogućujući sigurnu i korisnu integraciju ovih tehnologija u razne aspekte društva. Uspješno rješavanje problema halucinacija ključno je za daljnji razvoj i primjenu AI tehnologija, osiguravajući da njihova upotreba bude pouzdana i korisna za sve korisnike.

LITERATURA

[1] Hugging Face. URL: https://huggingface.co/. [Pristupljeno: 23. lipnja 2024.].

[2] OpenAI. URL: https://openai.com/. [Pristupljeno: 23. lipnja 2024.].

[3] Emotions of English tweet. URL: https://huggingface.co/spaces/trnt/twitter_emotions. [Pristupljeno: 25. lipnja 2024.].

[4] Gemma 1.1 7B (IT). URL: https://huggingface.co/google/gemma-1.1-7b-it. [Pristupljeno: 23. lipnja 2024.].

[5] Gradio documentation. URL: https://www.gradio.app/docs. [Pristupljeno: 19. lipnja 2024.].

[6] What is a chatbot?, . URL: https://www.ibm.com/topics/chatbots. [Pristupljeno: 22. lipnja 2024.].

[7] What are AI hallucinations?, . URL: https://www.ibm.com/topics/ai-hallucinations. [Pristupljeno: 23. lipnja 2024.].

[8] What is NLP (natural language processing)?, . URL: https://www.ibm.com/topics/natural-language-processing. [Pristupljeno: 20. lipnja 2024.].

[9] Introduction to Large Language Models. URL: https://developers.google.com/machine-learning/resources/intro-llms. [Pristupljeno: 21. lipnja 2024.].

[10] Meta Llama 3 8B Instruct. URL: https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct. [Pristupljeno: 23. lipnja 2024.].

[11] Mistral 7B Instruct v0.3. URL: https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3. [Pristupljeno: 23. lipnja 2024.].

[12] Nous Hermes 2 Mixtral 8x7B DPO. URL: https://huggingface.co/NousResearch/Nous-Hermes-2-Mixtral-8x7B-DPO. [Pristupljeno: 23. lipnja 2024.].

[13] OpenAI models. URL: https://platform.openai.com/docs/models. [Pristupljeno: 21. lipnja 2024.].

[14] SelfCheckGPT GitHub repozitorij. URL: https://github.com/potsawee/selfcheckgpt. [Pristupljeno: 20. lipnja 2024.].

[15] 32 ChatGPT Examples, Ideas & Use Cases, 2023. URL: https://gridfiti.com/chatgpt-examples/. [Pristupljeno: 25. lipnja 2024.].

[16] Introducing more enterprise-grade features for API customers, 2024. URL: https://openai.com/index/more-enterprise-grade-features-for-api-customers/. [Pristupljeno: 20. lipnja 2024.].

[17] Introducing improvements to the fine-tuning API and expanding our custom models program, 2024. URL: https://openai.com/index/introducing-improvements-to-the-fine-tuning-api-and-expanding-our-custom-models-program/. [Pristupljeno: 20. lipnja 2024.].

[18] Marc Caballé. Rule-Based Chatbots vs. AI Chatbots: Key Differences, 2023. URL: https://www.hubtype.com/blog/rule-based-chatbots-vs-ai-chatbots. [Pristupljeno: 22. lipnja 2024.].

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, i Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019. URL: https://arxiv.org/abs/1810.04805.

[20] Keith D. Foote. A Brief History of Large Language Models, 2023. URL: https://www.dataversity.net/a-brief-history-of-large-language-models/. [Pristupljeno: 21. lipnja 2024.].

[21] Keith D. Foote. A Brief History of Natural Language Processing, 2023. URL: https://www.dataversity.net/a-brief-history-of-natural-language-processing-nlp/. [Pristupljeno: 20. lipnja 2024.].

[22] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, i Ting Liu. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions, 2023. URL: https://arxiv.org/abs/2311.05232.

[23] Yichong Huang, Xiachong Feng, Xiaocheng Feng, i Bing Qin. The Factual Inconsistency Problem in Abstractive Text Summarization: A Survey, 2023. URL: https://arxiv.org/abs/2104.14839.

[24] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, i Pascale Fung. Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12):1–38, Ožujak 2023. ISSN 1557-7341. doi: 10.1145/3571730. URL: http://dx.doi.org/10.1145/3571730.

[25] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, i Jared Kaplan. Language models (mostly) know what they know, 2022. URL: https://arxiv.org/abs/2207.05221.

[26] Monika Karlović. 14 ways chatbots can elevate the healthcare experience, 2024. URL: https://www.infobip.com/blog/healthcare-ai-chatbot-examples. [Pristupljeno: 22. lipnja 2024.].

[27] Sachin Kumar. Detecting LLM Hallucinations: Strategies and Overview, 2024. URL: https://medium.com/@techsachin/detecting-llm-hallucinations-strategies-and-overview-57eea69e6a07. [Pristupljeno: 24. lipnja 2024.].

[28] Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, i Bill Dolan. A Token-level Reference-free Hallucination Detection Benchmark for Free-form Text Generation. U *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*,

stranice 6723–6737. Association for Computational Linguistics, Svibanj 2022. doi: 10.18653/v1/2022.acl-long.464. URL: https://aclanthology.org/2022.acl-long.464.

[29] Potsawee Manakul, Adian Liusie, i Mark J. F. Gales. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models, 2023. URL: https://arxiv.org/abs/2303.08896.

[30] Seyed Saeid Masoumzadeh. From Rule-Based Systems to Transformers: A Journey through the Evolution of Natural Language Processing, 2023. URL: https://medium.com/@masoumzadeh/from-rule-based-systems-to-transformers-a-journey-through-the-evolution-of-natural-language-9131915e06e1. [Pristupljeno: 21. lipnja 2024.].

[31] Wenyi Pi. Brief Introduction to the History of Large Language Models (LLMs), 2024. URL: https://medium.com/@researchgraph/brief-introduction-to-the-history-of-large-language-models-llms-3c2efa517112. [Pristupljeno: 21. lipnja 2024.].

[32] Alec Radford, Karthik Narasimhan, Tim Salimans, i Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[33] Asif Razzaq. Mistral AI Team Releases The Mistral-7B-Instruct-v0.3: An Instruct Fine-Tuned Version of the Mistral-7B-v0.3, 2024. URL: https://www.marktechpost.com/2024/05/22/mistral-ai-team-releases-the-mistral-7b-instruct-v0-3-an-instruct-fine-tuned-version-of-the-mistral-7b-v0-3/. [Pristupljeno: 23. lipnja 2024.].

[34] Deval Shah. The Beginner's Guide to Hallucinations in Large Language Models, 2023. URL: https://www.lakera.ai/blog/guide-to-hallucinations-in-large-language-models. [Pristupljeno: 23. lipnja 2024.].

[35] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, i Jason Weston. Retrieval Augmentation Reduces Hallucination in Conversation. U *Findings of the Association for Computational Linguistics: EMNLP 2021*, stranice 3784–3803. Association for Computational Linguistics, Studeni 2021. doi: 10.18653/v1/

2021.findings-emnlp.320. URL: https://aclanthology.org/2021.findings-emnlp.320.

[36] Niharika Singh. NousResearch Released Nous-Hermes-2-Mixtral-8x7B: An Open-Source LLM with SFT and DPO Versions, 2024. URL: https://www.marktechpost.com/2024/01/24/nousresearch-released-nous-hermes-2-mixtral-8x7b-an-open-source-llm-with-sft-and-dpo-versions/. [Pristupljeno: 23. lipnja 2024.].

[37] Anish Thapaliya. Google's BERT, 2021. URL: https://medium.com/@thapaliyanish123/google-bert-8e990b64f570. [Pristupljeno: 20. lipnja 2024.].

[38] S. M Towhidul Islam Tonmoy, S M Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, i Amitava Das. A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models, 2024. URL: https://arxiv.org/abs/2401.01313.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, i Illia Polosukhin. Attention Is All You Need, 2023. URL: https://arxiv.org/abs/1706.03762.

[40] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, i Denny Zhou. Self-consistency improves chain of thought reasoning in language models. U *The Eleventh International Conference on Learning Representations*, 2023. URL: https://openreview.net/forum?id=1PL1NIMMrw.

[41] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, i Ji-Rong Wen. A Survey of Large Language Models, 2023. URL: https://arxiv.org/abs/2303.18223.

Sustav za upravljanje halucinacijama velikih jezičnih modela

Sažetak

Halucinacije u velikim jezičnim modelima (LLM-ovima) narušavaju pouzdanost sadržaja generiranog pomoću umjetne inteligencije. Ovaj rad obuhvaća razvoj inovativnog sustava namijenjenog upravljanju halucinacijama u LLM-ovima putem interaktivne aplikacije. Sustav integrira više LLM-ova za generiranje i evaluaciju odgovora, koristeći mehanizme samoprovjere za detekciju haluciniranog sadržaja. Ključni dijelovi uključuju generiranje odgovora u stvarnom vremenu pomoću nekoliko LLM-ova, mehanizme samoprovjere za validaciju odgovora, te radni okvir za ocjenjivanje i kategorizaciju odgovora na temelju vjerojatnosti halucinacije. Aplikacija omogućuje korisnicima odabir između različitih modela te prilagodbu parametara poput temperature i praga detekcije, pružajući fleksibilnost u generiranju i evaluaciji odgovora.

Ključne riječi: obrada prirodnog jezika, veliki jezični modeli, chatbot, detekcija halucinacija, SelfCheckGPT

System for Managing Hallucinations in Large Language Models

Abstract

Hallucinations in large language models (LLMs) undermine the reliability of AI-generated content. This paper covers the development of an innovative system designed to manage hallucinations in LLMs through an interactive application. The system integrates multiple LLMs for generating and evaluating responses, employing self-check mechanisms for detecting hallucinated content. Key components include real-time response generation using several LLMs, self-check mechanisms for response validation, and a framework for scoring and categorizing responses based on hallucination likelihood. The application enables users to select from a range of models and adjust parameters such as temperature and detection threshold, providing flexibility in response generation and evaluation.

Keywords: natural language processing, large language models, chatbot, hallucination detection, SelfCheckGPT