Fakultet primijenjene matematike i informatike

Tech-Math AI Agent

Kolegij: Obrada prirodnog jezika tehnikama dubinskog učenja

Student:

Zoltan Palinkaš

Osijek, 2026

Sadržaj

Tablica slika

1. Uvod

U posljednjem desetljeću svjedočimo ubrzanom razvoju velikih jezičnih modela (Large Language Models – LLM), koji su postali ključni alat u području umjetne inteligencije. Modeli poput GPT-a, Mistrala i sličnih sustava pokazali su iznimnu sposobnost razumijevanja prirodnog jezika, generiranja tekstova i asistiranja korisnicima u širokom spektru zadataka. Unatoč tome, njihova primjena u znanstvenim i tehničkim disciplinama još uvijek nailazi na određena ograničenja.

Dva osnovna problema koja se često pojavljuju kod LLM sustava su fenomen halucinacija te ograničena pouzdanost u matematičkom i numeričkom računanju. Halucinacije predstavljaju situaciju u kojoj model generira uvjerljivo formulirane, ali netočne ili izmišljene informacije. S druge strane, matematički izračuni zahtijevaju visoku razinu preciznosti koju modeli temeljeni na predviđanju tokena često ne mogu garantirati.

Kako bi se riješili navedeni problemi, u ovom radu predstavljen je sustav nazvan **Tech-Math AI Agent**. Riječ je o hibridnom sustavu koji kombinira nekoliko modernih pristupa umjetne inteligencije. Sustav koristi Retrieval-Augmented Generation (RAG) za povećanje faktografske točnosti odgovora, dok Program-Aided Language (PAL) omogućuje generiranje i izvršavanje Python koda radi postizanja matematičke preciznosti.

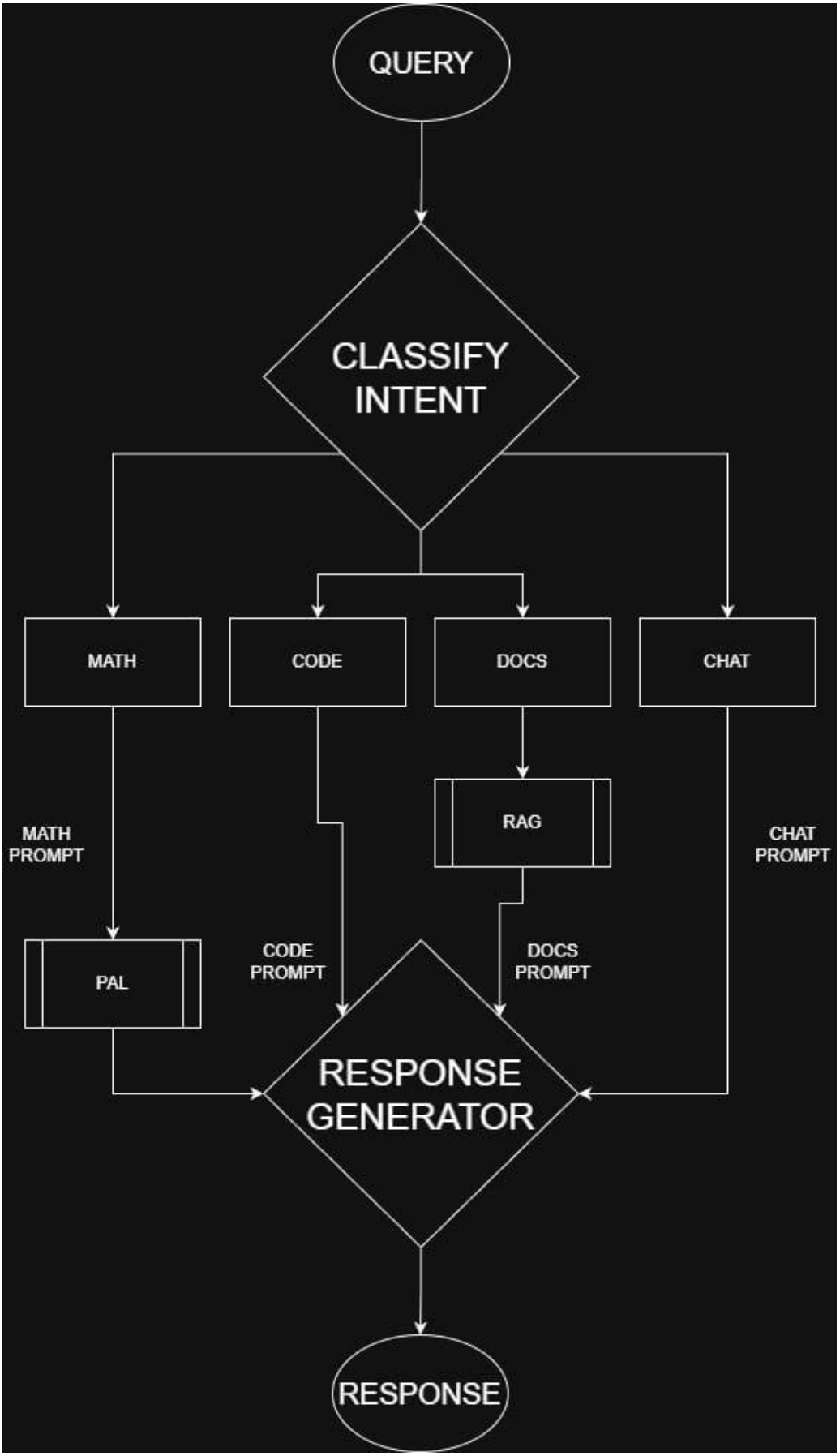
Cilj ovog rada je analizirati arhitekturu sustava, opisati njegovu implementaciju te evaluirati njegovu učinkovitost u rješavanju matematičkih i dokumentacijskih zadataka iz područja numeričke matematike.

2. Arhitektura sustava

Središnji dio sustava **Tech-Math AI Agent** temelji se na modularnoj arhitekturi koja omogućuje fleksibilno procesiranje korisničkih upita. Sustav koristi cjevovod (pipeline) koji omogućuje analizu upita i usmjeravanje prema odgovarajućem modulu.

Glavna komponenta sustava je Intent Classifier, čija je zadaća prepoznati vrstu korisničkog zahtjeva. Ovakav pristup omogućuje optimizaciju resursa jer se izbjegava nepotrebno korištenje kompleksnih modula kada oni nisu potrebni. Nakon klasifikacije, upit se prosljeđuje odgovarajućem podsustavu koji generira odgovor.

Ovakva arhitektura povećava skalabilnost sustava, omogućuje lakše održavanje te poboljšava točnost rezultata. Modularni pristup također omogućuje jednostavnu integraciju novih funkcionalnosti u budućnosti.



Slika 1: Arhitektura

2.1. Klasifikacija namjere (Few-Shot Prompting)

Za klasifikaciju korisničkih upita koristi se metoda **In-Context Learning**, koja omogućuje modelu da na temelju primjera prepozna kategoriju upita bez dodatnog treniranja modela. Sustav klasificira upite u četiri osnovne kategorije:

- **MATH** – matematički izračuni
- **DOCS** – teorijska pitanja vezana uz dokumentaciju
- **CODE** – generiranje ili analiza programskog koda
- **CHAT** – opći razgovorni upiti

Ovaj pristup omogućuje optimizaciju rada sustava jer, primjerice, RAG modul nije potrebno koristiti za jednostavne konverzacijske upite. Time se smanjuje potrošnja računalnih resursa i ubrzava generiranje odgovora.

Few-shot prompting dodatno poboljšava točnost klasifikacije jer model dobiva primjere ispravnih klasifikacija. Ipak, rezultati pokazuju da model povremeno pogrešno klasificira složenije upite koji kombiniraju teorijske i praktične elemente.

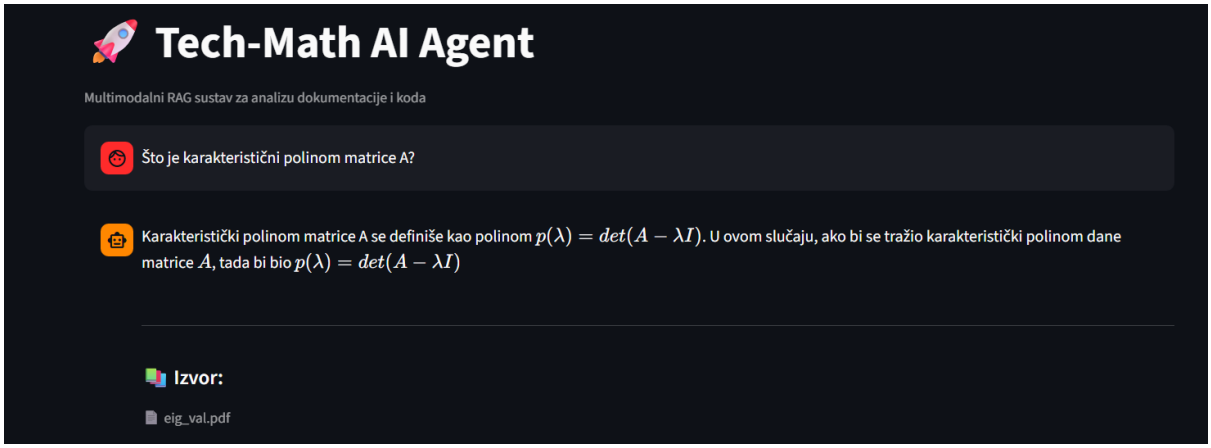
2.2. RAG Modul (Obrada dokumenata)

Retrieval-Augmented Generation predstavlja tehniku koja kombinira dohvaćanje relevantnih informacija iz baze podataka s generiranjem odgovora pomoću jezičnog modela.

Za pretvorbu tekstualnih podataka u numeričke reprezentacije koristi se embedding model **intfloat/multilingual-e5-small**, koji je optimiziran za višejezične zadatke i pruža dobre rezultate za hrvatski jezik.

Pohrana embeddinga realizirana je pomoću **FAISS vektorske baze**, koja omogućuje brzo pretraživanje velikog broja tekstualnih segmenata. Dokumenti se dijele na manje dijelove (chunkove) duljine 500 znakova uz preklapanje od 180 znakova. Preklapanje je važno jer omogućuje očuvanje konteksta matematičkih formula i definicija koje se često protežu kroz više rečenica.

Integracijom RAG-a sustav može generirati odgovore koji su temeljeni na stvarnim nastavnim materijalima, čime se značajno smanjuje rizik od halucinacija.



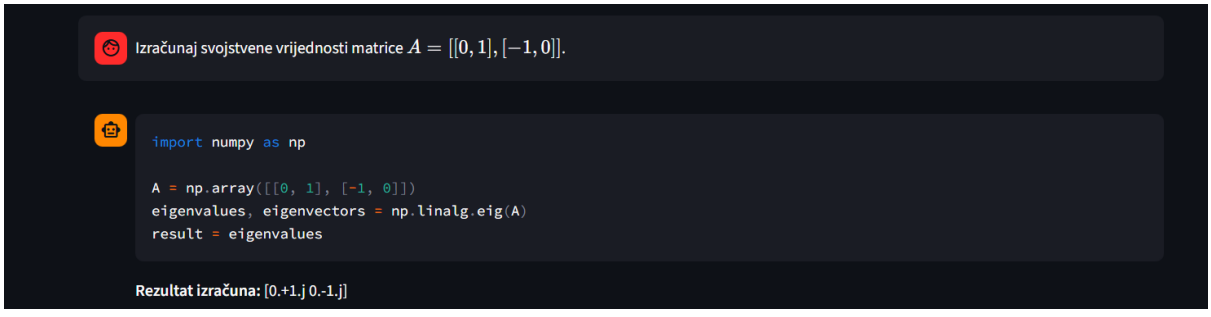
Slika 2: Odgovor uz pomoć RAG-a

2.3. PAL Modul (Matematičko izvršavanje)

Program-Aided Language modul predstavlja ključnu komponentu sustava za rješavanje matematičkih problema. Umjesto da model sam izvodi numeričke izračune, on generira Python kod koji se zatim izvršava u lokalnom okruženju.

PAL modul koristi biblioteku NumPy, koja omogućuje učinkovito izvođenje matričnih operacija i numeričkih metoda. Ovakav pristup osigurava visoku razinu točnosti jer se matematičke operacije izvršavaju deterministički, za razliku od probabilističke prirode LLM modela.

Osim povećanja točnosti, PAL omogućuje transparentnost rada sustava jer korisnik može vidjeti generirani kod i provjeriti postupak izračuna. Time se sustav približava znanstvenim standardima reproducibilnosti rezultata.



Slika 3: Odgovor uz pomoć računanja

3. Tehnička implementacija

Sustav je implementiran lokalno s ciljem zaštite privatnosti podataka i smanjenja ovisnosti o vanjskim servisima. Kao temeljni model koristi se **Mistral-7B-Instruct-v0.3**, koji je kvantiziran na 4-bitnu preciznost pomoću biblioteke bitsandbytes. Kvantizacija omogućuje rad modela na grafičkim karticama s ograničenom količinom memorije (6 GB VRAM-a), uz minimalan gubitak performansi.

Isti model koristi se u dvije ključne funkcionalnosti sustava. Prva funkcionalnost odnosi se na **klasifikaciju namjere korisničkog upita (Intent Classification)**. U tom slučaju model koristi *few-shot prompting* pristup, pri čemu na temelju zadanih primjera određuje pripada li upit kategoriji MATH, DOCS, CODE ili CHAT. Ova faza predstavlja početni korak obrade upita i omogućuje sustavu odabir odgovarajuće strategije generiranja odgovora.

Nakon klasifikacije, model se koristi za **generiranje tekstualnih odgovora**, gdje interpretira korisničke upite, integrira informacije dobivene putem RAG modula te generira objašnjenja, matematičke postupke ili programski kod. Ovakav pristup omogućuje fleksibilno prilagođavanje odgovora ovisno o vrsti korisničkog zahtjeva te poboljšava ukupnu točnost i relevantnost generiranih rezultata.

Nakon klasifikacije, sustav dinamički prilagođava način komunikacije s modelom. Ovisno o prepoznatoj namjeri korisnika, generira se specifičan prompt koji usmjerava model prema željenom tipu odgovora. Primjerice, matematički upiti koriste strukturirane promptove koji potiču generiranje Python koda za PAL modul, dok DOCS upiti koriste promptove optimizirane za integraciju informacija iz RAG sustava. Ovakav pristup omogućuje povećanje točnosti odgovora, smanjenje halucinacija te poboljšanje ukupne pouzdanosti sustava.

Korisničko sučelje razvijeno je pomoću frameworka Streamlit, koji omogućuje jednostavno učitavanje dokumenata, interakciju s korisnikom i prikaz matematičkih izraza u LaTeX formatu. Sučelje omogućuje dinamičko upravljanje dokumentima koji se koriste u RAG modulu te pregled generiranog programskog koda i rezultata izvođenja. Time je sustav prilagođen obrazovnom okruženju i omogućuje intuitivno korištenje.

4. Eksperimentalna usporedba i Evaluacija

Evaluacija sustava provedena je na skupu od 15 testnih pitanja koja pokrivaju četiri kategorije korisničkih upita: matematičke zadatke, teorijska pitanja iz dokumentacije, generiranje programskog koda i opće konverzacijske upite.

Rezultati testiranja pokazuju da je sustav generirao točne odgovore u 10 od ukupno 15 slučajeva, što odgovara ukupnoj točnosti od 66,7%. Analiza pokazuje da sustav postiže najbolje rezultate kod deterministčkih matematičkih zadataka koji se mogu riješiti korištenjem PAL modula i NumPy biblioteke.

Kod klasifikacije namjere korisničkog upita sustav je ostvario višu razinu točnosti. Intent classifier ispravno je klasificirao 12 od 15 upita, što predstavlja 80% točnosti klasifikacije. Pogreške su zabilježene u 3 slučaja, uglavnom kod upita koji kombiniraju teorijske i praktične elemente.

Najveća pouzdanost sustava primijećena je u kategoriji MATH, gdje je 4 od 6 matematičkih zadataka riješeno potpuno točno. Kod DOCS upita sustav je dao 3 potpuno točna i 1 djelomično točan odgovor. U CODE kategoriji svi generirani programski zadaci bili su funkcionalno točni, iako su u dva slučaja bili pogrešno klasificirani kao matematički upiti. CHAT kategorija pokazala je najveću varijabilnost rezultata, s 2 točna i 2 netočna odgovora.

Dobiveni rezultati potvrđuju da kombinacija RAG i PAL pristupa značajno poboljšava pouzdanost odgovora u usporedbi s klasičnim LLM sustavima koji se oslanjaju isključivo na generiranje teksta.

#	Upit	Očekivani intent	Generirani intent	Točan odgovor (DA/NE)	Napomena
1.	Što je karakteristični polinom matrice A?	DOCS	DOCS	DA	
2.	Kako glasi formula za Jacobijevu iteraciju u matričnom obliku?	DOCS	MATH	NE	
3.	Objasni što je Hessenbergova forma matrice i kako se do nje dolazi (Householderovi reflektori).	DOCS	DOCS	DA	
4.	Koju procjenu pogreške mora zadovoljavati	DOCS	DOCS	DA*	Djelomično točan

	Jacobijeva metoda prema priloženim zadacima?				
5.	Izračunaj svojstvene vrijednosti matrice A $= \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$.	MATH	MATH	DA	
6.	Za sustav Ax=b gdje je A $= \begin{pmatrix} 10 & 1 \\ 1 & 10 \end{pmatrix}$ i b = $\begin{pmatrix} -8 \\ 0 \end{pmatrix}$, izračunaj prvu aproksimaciju $x^{(1)}$ Jacobijevom metodom uz $x^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.	MATH	MATH	NE	Generirao je krivi kod
7.	Ako je A = $\begin{pmatrix} 6 & 1 & 0 \\ 1 & 5 & 1 \\ 0 & 1 & 4 \end{pmatrix}$ izračunaj determinantu te matrice.	MATH	MATH	DA	
8.	Provjeri jesu li vektori $r^{(0)} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ i $r^{(1)} = \begin{pmatrix} -2 \\ 1 \end{pmatrix}$ međusobno okomiti (skalarni produkt).	MATH	MATH	DA	
9.	Odredi broj iteracija k potreban da pogreška Jacobijeve metode padne ispod 10^{-3} ako je $\|C\| = 0.2$ i $\|x^{(1)}-x^{(0)}\| = 1$.	MATH	MATH	NE	Računao je aproksimaciju umjesto broja iteracija
10.	Napiši Python funkciju koja prima matricu i vraća njezinu transponiranu matricu koristeći list comprehension.	CODE	CODE	DA	
11.	Generiraj kod koji simulira jednu iteraciju Gauss-	CODE	MATH	DA	Krivo je klasificirao intent, ali je

	Seidelove metode za sustav 3x3.				generirao dobar kod
12.	Napiši skriptu koja provjerava je li zadana matrica simetrična i pozitivno definitna.	CODE	MATH	DA	Krivi intent, ali dobar kod
13.	Tko si ti, koji model koristiš u pozadini i tko te programirao?	CHAT	DOCS	NE	Halucinirao je odgovor
14.	Kakvo je vrijeme danas u Osijeku?	CHAT	CHAT	NE	
15.	Koje je boje nebo?	CHAT	CHAT	DA	

5. Zaključak

Rezultati ovog istraživanja pokazuju da sustav Tech-Math AI Agent može uspješno kombinirati prednosti velikih jezičnih modela s deterministčkim numeričkim metodama. Integracija RAG modula omogućuje visoku faktografsku točnost, dok PAL modul osigurava matematičku pouzdanost.

Istraživanje potvrđuje da i relativno mali modeli, poput 7B modela, mogu postići visoku razinu funkcionalnosti kada su integrirani s odgovarajućim alatima i arhitekturom sustava. Time se otvara mogućnost razvoja lokalnih AI rješenja koja su dostupna obrazovnim institucijama i istraživačima.

Budući rad usmjerit će se na implementaciju mehanizama samoispravljanja (Self-Correction), poboljšanje klasifikacije upita te proširenje sustava na dodatne matematičke metode i znanstvene discipline.