UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

SPECIALIZAREA INFORMATICĂ

Lucrare de licență

CORECTAREA AUTOMATA CU AJUTORUL LLM-URILOR : STUDIU DE CAZ - BACALAUREAT INFORMATICĂ

Absolvent
Firca Liviu Nicolae

Coordonator științific
Marius Adrian Dumitran

București, Iunie 2025

Rezumat

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce vitae eros sit amet sem ornare varius. Duis eget felis eget risus posuere luctus. Integer odio metus, eleifend at nunc vitae, rutrum fermentum leo. Quisque rutrum vitae risus nec porta. Nunc eu orci euismod, ornare risus at, accumsan augue. Ut tincidunt pharetra convallis. Maecenas ut pretium ex. Morbi tellus dui, viverra quis augue at, tincidunt hendrerit orci. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam quis sollicitudin nunc. Sed sollicitudin purus dapibus mi fringilla, nec tincidunt nunc eleifend. Nam ut molestie erat. Integer eros dolor, viverra quis massa at, auctor.

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce vitae eros sit amet sem ornare varius. Duis eget felis eget risus posuere luctus. Integer odio metus, eleifend at nunc vitae, rutrum fermentum leo. Quisque rutrum vitae risus nec porta. Nunc eu orci euismod, ornare risus at, accumsan augue. Ut tincidunt pharetra convallis. Maecenas ut pretium ex. Morbi tellus dui, viverra quis augue at, tincidunt hendrerit orci. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam quis sollicitudin nunc. Sed sollicitudin purus dapibus mi fringilla, nec tincidunt nunc eleifend. Nam ut molestie erat. Integer eros dolor, viverra quis massa at, auctor.

Cuprins

Capitolul 1

Introducere

Bacalaureatul este unul dintre cele mai importante examene din invatamant. El permite accesul la studii superioare. Pe baza notelor primite, si accesul anumitor institutii de invatamant superior de top. Din acest motiv obiectivitatea si corectarea cat se poate de corecta este necesara.

Capitolul 2

Preliminarii

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean dignissim metus justo, nec pharetra mauris tincidunt id. Praesent semper turpis quis faucibus pulvinar. Fusce ut justo nisi. Praesent vehicula blandit erat, sed dignissim justo bibendum lobortis. Vivamus fringilla, elit at pulvinar imperdiet, dui elit lobortis sapien, et vehicula urna ex et velit. Sed efficitur, neque sed egestas lobortis, diam leo pellentesque sem, nec gravida est nunc efficitur orci. Maecenas bibendum pharetra sapien, quis porttitor justo vestibulum a. Nunc tempus erat sed augue blandit, eu scelerisque dolor lobortis. Vestibulum quam arcu, malesuada quis felis eleifend, iaculis gravida massa. Vestibulum lacinia arcu nec risus laoreet porttitor. Quisque ut nisl consequat, sodales nisl id, pretium neque. Interdum et malesuada fames ac ante ipsum primis in faucibus. Vivamus tincidunt, ligula in fringilla laoreet, urna risus efficitur orci, eget pellentesque metus magna interdum lorem.

Quisque et ligula erat. Aliquam eget fringilla tortor. Nulla maximus, massa ac consectetur fringilla, ante velit porttitor justo, et euismod magna eros consequat nunc. Praesent lacinia nulla dui, lobortis mattis odio ullamcorper et. Proin vehicula massa in efficitur eleifend. Mauris ornare mi ac rutrum porta. Morbi sed magna vel sem egestas malesuada ac at turpis. Aenean eleifend pharetra massa, eget porta ligula eleifend ultricies. Donec vel imperdiet leo. Nullam commodo metus metus, sed semper velit ornare posuere.

Aliquam erat volutpat. Pellentesque vulputate massa sed semper lacinia. Duis hendrerit dolor et blandit malesuada. Curabitur posuere tellus at lacinia scelerisque. Donec hendrerit semper ullamcorper. Morbi ut semper metus. Aliquam pharetra sagittis dolor, non pulvinar purus euismod non. Phasellus condimentum a est tristique convallis. Nam ac mauris bibendum, lobortis ex at, rhoncus nibh. Fusce neque nisi, dignissim porttitor lorem sed, lobortis vehicula felis.

Capitolul 3

Colectarea si prelucrarea datelor

3.1 Colectarea datelor

Datele au fost generate de studenti in aplicatia teams a facultatii. Modul in care au lucrat a fost urmatorul: li s-a asignat un subiect de bac la informatica din 2020 pana in 2024 si au trebuit sa isi aleaga un numar aleatoriu dintre 5 si 10. Dupa acea au fost instruiti sa faca greseli astfel incat nota lor sa corespunda cu numarul ales. Pentru a facilita corectarea lucrarilor, studentii au si trebuit sa indice greselile facute, dar si nota pe care si-au asignat-o. Am primit in total 42 de lucrari, dintre care au fost filtrate 9 lucrari pentru ca aveau date de proasta calitate. Problemele pe care le-am intalnit aici erau multiple

- Niste studenti pur si simplu nu au ales subiectul care trebuie (sesiunea gresita, profilul stiintele naturii in loc de Matematica-Informatica)
- Niste studenti au trimis in formatul gresit (fisier .docx a interferat cu sitaxa colului, fisiere excel in loc de .txt)
- Niste studenti pur si simplu nu erau programatori puternici, si nu s-au verificat destul , deci cand a fost efectiv corectata lucrarea au avut mai multe greseli decat au raportat. Acest lucru este problematic pentru ca incetineste mult corectarea lucrarilor. Totusi studentii erau din anul intai si aceste greseli au fost cel mai probabil neintentionate

Pentru a evita aceste probleme puteam face 2 lucrui:

- Sa creem un formular pentru submitere
- Sa mobilizam studenti cu mai multa experienta in programare

3.2 Prelucrarea datelor

Datele folosite au fost nu numai lucrarile studentilor, dar si subiectele si barelmele fiecarui an. Acestea au trebuit sa fie rescrise pentru a avea o structura mai adapatata pentru LLM-uri. In plus de acestea, au fost rescrise intr-un limbaj mai simplu si usor de inteles. Am filtrat o a doua oara lucrarile studentilor pana a avea 23 de lucrari in total. Acest lucru a fost facut din 2 motive : a limita costurile lucrarii (fiecare query costa relativ mult) si a permite testarea a mai multor LLM-uri, dar si pentru a avea date de calitate. Lucrarile au fost corectate de autor, cu atentie sporita la barem dar si la potentialele erori facute de elev (tot continutul lucrarilor a trebuit recorectat, pentru a garanta o nota cat mai apropiata de realitate)

Capitolul 4

Corectarea automata si feedback automat

4.1 Corectarea automata

Una dintre premisele lucrarii este ca daca LLM-ul are o distanta mai mica de 0.5 puncte din 10, atunci ar putea teoretic inlocui un corector uman. Ne intrebam in primul rand daca cea mai directa aboradare (se da toata lucrarea, tot baremul, si tot subiectul si este instruit sa noteze lucrarea pe baza baremului) produce rezultate bune. Pentru acest lucru o sa folosim GPT 4.1 din 2 motive: Nu este un thinking model (vrem sa avem abordarea cat mai simpla), si este un model recent pentru studiu. Eroarea folosita este mean absolute error (MAE), dar se ia in calcul si cel mai prost rezultat CMPR , cand distanta intre nota reala si cea asignata a fost cea mai mare

Modelul	**MAE**	**CMPR**
GPT 4.1	4.78	18

Tabela 4.1: Prima aboradare, punctaj acordat din 100 puncte

Cum putea vedea modelul se incadreaza tehnic in limita superioara impusa (5 puncte din 100). Totusi, este foarte aproape de aceasta limita, si CMPR este foarte mare.

In plus analizand justificarea acordata a notelor puteam observa mai multe lucruri in neregula:

- Uneori nu corecteaza pe baza baremului, de exemplu daca vede un program scris ”foarte lenes”, el ii va da 0 puncte chiar daca baremul generos de la bac i-ar acorda 5 puncte
- Uneori LLM-ul a rescris codul cu intentia de al puncta (lucru care se intampla daca textul dat este foarte lung), dar a omis o parte a coduli care a judecat-o irelevanta. Dar dupa ce a trecut prin cerintele baremului, a depunctat studentul pentru ca

lipseste partea a codului care nu a vrut sa o copieze. Cred ca lucrul acesta are 2 cauze simultane : un LLM nu se poate ”gandii” la tot textul in acelasi timp, deci nu isi da seama de eroarea comisa. In plus, planificarea in avans a fost clar proasta.

- Uneori nu si-a dat seama ca raspunsul corect se afla in barem, deci a incercat sa gaseasca singur solutia, dar nu a fost correct. Deci corectarea a fost si ea gresita

Multe dintre neintelegerile sale au fost din cauza contextului prea mare, care l-a facut sa ignore baremul sau sa corecteze gresit. Din cauza asta avem o a doua abordare: ne dam seama ca fiecare intrebare de la bac este independenta, deci se imparte fiecare lucrare in 7 ”lucrari”, si baremul si subiectul sunt si ele impartite cum trebuie. In plus de asta folosim mai multe modele, inclusiv si thinking models. Din curiozitate, am inclus si GPT 4, pentru a vedea cat de important este factorul ca modelelel sunt noi. Folosind aceleasi metrici:

Modelul	**MAE**	**CMPR**
GPT 4.1	3.63	12
GPT 4	11.06	25.5
Gemini 2.5 flash	3.67	11
Gemini 2.5 pro	3.09	11
Deepseek v3	5.8	14
Deepseek r1	4.78	14

Tabela 4.2: A doua abordare, punctaj acordat din 100 puncte

Putem remarca imediat ca GPT 4 este un model foarte prost in comparare cu ceilalti, deci confirma idea ca LLM-urile s-au ameliorat foarte mult in anii recenti si nu mai este cazul de a folosi unul care nu este recent. Un LLM recent este mult mai performant si de multe ori mult mai ieftin de folosit. Un alt lucru care iese la iveala este ca resursele alocate inferentei este crucial: exista o diferenta significativa intre Gemini 2.5 flash si Gemini 2.5 pro de 0.5 puncte. In plus de asta exista o diferenta significativa intre Deepseek v3 si Deepseek r1 de 1 punct, iarasi sustinand idea ca mai multe resurse alocate la inferente rezulta la predictii mai bune. Deci se pot distinge 4 factori cruciali in a determina performanta predictiilor:

- Resursele alocate inferentei (putere de calcul $\times$timp)
- Cat de recent este LLM-ul folosit (cu atat este mai recent, cu atat mai bine)

Putem remarca si faptul ca GPT 4.1 si Gemini 2.5 pro au avut erori foarte mici: 3.6 si 3. Acest lucru este destul de impresionant pentru ca exista o parte subiectiva in a interpreta baremul, si rezultatul cel mai bun (3) este cu mult sub limita superioara impusa. CMPR este totusi destul de mare pentru ambele modele. Doua greseli pe care am putut sa le identific este ca :

- O data un student a facut o greseala in logica codului, si Gemini 2.5 pro a spus ca algoritmul este gresit, cand era clar doar o mica greseala de logica
- O data un student nu a respectat cerinta, dar GPT 4.1 i-a acordat toate punctele, pentru ca nu si-a dat seama ca structurile **repeta ... pana cand ...** si **pana cand ... repeta ...** sunt foarte diferite in contextul cerintei : trebuia ca studentul sa inlocuiasca a doua structura in pseudocod cu prima, pentru a arata ca a inteles cum se pot echivala algoritmic ambele. Studentul nu a folosit structura care trebuie, dar a primit totusi toate punctele.

Ca solutie la aceste probleme, ar putea fi folosit un barem mai explicit, mai redundant, sau poate chiar si fine-tuning.

4.2 Feedback automat

Vrem sa ne interesam daca LLM-urile pot sa analizeze cum trebuie greselile studentilor, si sa propuna feedback relevant profesorului (ce parte din materie nu a fost bine inteleasa de catre studenti, si trebuie predata mai bine). Pentru asta, luam cele top 2 modele in performanta, si le testam. Aceasta parte este mult mai subiectiva decat cea precedenta, pentru ca este greu sa cuantifici calitatea unui feedback. Probelemele comune, identificate de mine sunt :

- sintaxa lui C++
- Algoritmica nu este inteleasa bine
- Intrebarile nu sunt citite bine, si de multe ori studentii s-au aruncat in a scrie o solutie fara sa se gandeasca
- I/O este implementat prost
- Prelucrarea stringurilor (char[ ]) nu este bine inteleasa
- Studentii pierd foarte mult puncte la intrebarile cu raspunsuri multiple

Problemele comune, identificate de GPT 4.1 sunt:

1. Algoritmica nu este inteleasa bine
2. ~~Studentii au probleme cand lucreaza cu matrici~~ (acest lucru este categoric fals)
3. Prelucrarea stringurilor (char[ ]) nu este bine inteleasa
4. I/O este implementat prost
5. Parametrii referinta (int f(&n)) nu sunt bine folositi

6. Initializarile sunt prost facute

Deci LLM-ul a identificat niste probleme foarte importante, unele care nu au fost identificate de mine. Totusi (2) este un feedback gresit. Cea ce lipseste in acest feedback este ca sintaxa nu a fost bine inteleasa, si ca studentii pierd prea multe puncte la intrebarile cu raspuns multiplu.

Cand vine vorba de gemini 2.5, a identificat toate problemele pe care le-am vazut, mai putin problema cu intrebarile cu raspuns multiplu. Nu am primit feedback gresit din partea lui gemini 2.5 .

Pentru a completa partea de feedback, le-am cerut LLM-urilor sa genereze 10 probleme, pentru ca studentii sa se poata antrena cu ele. Din cele 10 generate de GPT 4.1, 3 au fost irelevante, si dintre cele 10 generate de gemini, numai 1 a fost irelevanta.

Putem trage concluzia ca :

- Se pare ca o performanta mai buna in MAE si CMPR se transpune intr-o performanta mai buna cand vine vorba de feedback
- Gemini 2.5 este cel mai bune model, si la corectare automata, dar si la feedback

Totusi, ambele LLM-uri au dat feedback relevant.

Capitolul 5

Concluzii