COMPETIȚIA DE PROIECTE DE CERCETARE A ACADEMIEI OAMENILOR DE ȘTIINȚĂ DIN ROMÂNIA DESTINATĂ TINERILOR CERCETĂTORI „AOȘR-TEAMS-IV” EDIȚIA 2025-2026 „TRANSFORMAREA DIGITALĂ ÎN ȘTIINȚE”

Platformă web pentru generarea automată de întrebări (QuizTools)

Raport 2

Director de proiect: Conf. dr. ing. Ștefan Rușeți
Membru: Sl. dr. ing. Răzvan Păroiu
Membru: As. drd. ing. Andreea Duțulescu

Cuprins

1 Introducere

Generarea automată de întrebări reprezintă o abordare scalabilă a evaluării educaționale, permițând feedback rapid și sprijinind înțelegerea textelor citite. Cu toate acestea, producerea de întrebări care se aliniază cu obiective diverse continuă să prezinte provocări tehnice. În consecință, sarcina generării de întrebări controlabile prin atribute implică producerea de întrebări în limbaj natural care respectă atribute predefinite, cum ar fi dificultatea, focusul sau tipologia. Un astfel de control este din ce în ce mai important în aplicațiile educaționale.

În ciuda interesului crescând pentru acest domeniu, abordarea predominantă rămâne antrenarea clasică supervizată, în care modelele sunt antrenate pe seturi de date adnotate cu etichete explicite ale atributelor. Un alt obstacol este lipsa seturilor de date adnotate cu atribute multiple, ceea ce limitează semnificativ capacitatea modelelor de a generaliza către noi combinații de atribute. Acest lucru necesită adesea strategii de augmentare a datelor sau metode proxy. Deși câteva modele (Dong et al., 2023; Tu et al., 2024) au fost propuse pentru a facilita generarea controlabilă, acestea sunt concepute sub presupunerea că doar un subset limitat de atribute este important în momentul generării. Prin urmare, acestea sunt insuficiente în scenariile care necesită un control complet al atributelor. Din câte știm, există o lipsă de soluții complete capabile să impună un control cuprinzător și simultan asupra tuturor atributelor relevante în generarea de întrebări.

Pentru a aborda aceste provocări, propunem optimizarea direcționată a raportului de șanse - Odda-Ratio Steerable Optimization (ORSO), o paradigmă de antrenare care îmbunătățește atenția modelului pe atribute. Spre deosebire de antrenarea supervizată clasică, unde modelul este încurajat doar să maximizeze probabilitatea unui rezultat având în vedere o intrare condiționată de atribute, ORSO penalizează explicit modelul pentru producerea aceluiași rezultat în condiții de configurații de atribute diferite și incompatibile. Aceasta vizează direct problema în care modelele tind să ia în considerare doar în linii mari valorile atributelor.

Ne bazăm pe progresele recente în învățarea bazată pe preferințe, dar în loc să folosim perechi de preferințe selectate de oameni, ORSO valorifică perturbațiile la nivel de intrare. Mai exact, o intrare pozitivă care conține valorile corecte ale atributelor este asociată pentru fiecare instanță de antrenare cu o intrare negativă în care atributele sunt corupte intenționat, menținând în același timp ieșirea țintă fixă. Obiectivul de antrenare încurajează apoi modelul să atribuie o probabilitate mai mare ieșirii condiționate de configurația corectă a atributelor decât celei perturbate. În această lucrare, susținem că ORSO obține o sensibilitate superioară a atributelor și o fidelitate de generare pentru mai multe variabile de control, depășind metodele convenționale supervizate.

2 Lucrări conexe

2.1 Generare controlabilă

Li & Zhang (2024) au introdus un model de generare controlabilă de întrebări în doi pași, combinând planificarea răspunsurilor și sinteza întrebărilor. În primul rând, un model mare de limbaj (LLM) generează un plan de răspuns structurat dintr-un context dat, ghidat de semnale de control. Planul constă în unități de informații cheie sau intervale de răspunsuri candidate. În a doua etapă, contextul, planul și promptul bazat pe control au fost utilizate ca date de intrare pentru un model de generare a întrebărilor pentru a produce perechi întrebare-răspuns aliniate. Deși abordarea a atins o calitate puternică a generării, autorii nu și-au lansat modelele pentru evaluare deschisă, ceea ce limitează reproductibilitatea.

Tu et al. (2024) au introdus un cadru de decodare constrânsă pentru LLM-uri pentru a ghida generarea de text pentru satisfacerea constrângerilor arbitrare (de exemplu, includerea lexicală, evitarea toxicității). La fiecare pas de decodare, următoarele cuvinte candidate sunt reclasificate nu numai în funcție de scorurile lor standard de probabilitate, ci și de o estimare auxiliară a satisfacerii constrângerilor. Rezultatele experimentale au indicat că metoda lor a depășit decodarea standard greedy și beam search, dar cu un cost computațional ridicat.

Dong et al. (2023) au introdus SteerLM. Spre deosebire de RLHF, care se bazează pe o rețea complexă de antrenare online, SteerLM acceptă condiționare multi-atribut (de exemplu, gradul de ajutor, umor, toxicitate) în momentul inferenței. Antrenarea a fost realizat prin adăugarea de valori discrete ale atributelor la promptul de intrare, condiționând generarea modelului de semnale explicite. Acest mecanism de condiționare este aplicat în timpul unei antrenări supervizate clasice, utilizând atât atribute adnotate de om, cât și atribute prezise automat. Pentru a îmbunătăți performanța, autorii au generat date suplimentare de antrenare prin eșantionarea ieșirilor modelului. Evaluările empirice au susținut că această metodă a depășit performanțele de referință ale RLHF atât în evaluările automate, cât și în cele umane. Datorită performanței sale puternice și a controlabilității atributelor, SteerLM constituie o bază deosebit de relevantă pentru evaluarea strategiilor alternative de aliniere.

Guo et al. (2024) au propus o modificare a Direct Preference Optimization (DPO; Rafailov et al., 2023) pentru a alinia LLM-urile cu mai multe obiective concurente (de exemplu, utilitatea, toxicitatea, onestitatea). Recunoscând că aceste obiective nu pot fi maximizate simultan, metoda a căutat soluții Pareto-optimale prin condiționarea generării de preferințele specificate de utilizator, similar cu SteerLM . În antrenare preferințelor, recompensele au fost calculate pe baza abaterii dintre rezultatele generate și preferințele țintă, rezultatul cu recompensă mai mare fiind tratat ca eșantion

preferat. Rezultatele au arătat o aliniere îmbunătățită față de DPO-ul inițial în contexte cu obiective multiple.

2.2 Metode de aliniere și antrenare semi-supervizată

În domeniu au fost propuse multiple tehnici de aliniere a preferințelor (Ouyang et al., 2022; Rafailov et al., 2023; Gheshlaghi Azar et al., 2024). ORPO (Odds-Ratio Preference Optimization) a fost propusă de Hong et al. (2024) pentru alinierea cu preferințele oamenilor, pentru a elimina necesitatea unui model de referință. Metoda a combinat probabilitatea logaritmică negativă (NLL) standard cu o penalizare a raportului de șanse pentru răspunsurile alese și respinse, pe baza probabilităților lor de generare. Evaluările empirice pe diverse dimensiuni de model și repere au arătat că ORPO a depășit adesea paradigmele RLHF sau DPO. Autorii au furnizat, de asemenea, dovezi teoretice pentru utilizarea raportului de șanse în locul raportului de probabilități în modelarea preferințelor, subliniind proprietățile sale.

Aceste tehnici de aliniere, precum și antrenarea supervizată clasică, au servit drept bază pentru paradigmele de antrenare semi-supervizată la completarea seturilor de date cu eșantioane sintetice generate de modelele în sine. Au fost propuse multiple studii iterative de antrenare semi-supervizată.

Jung et al. (2024) au propus un cadru semi-supervizat pentru bootstrapping-ul etichetelor folosind LLM-uri. Inițial, un set mic de date etichetate a fost utilizat pentru a solicita unui LLM să adnoteze eșantioane neetichetate. Doar predicțiile modelului cu încredere ridicată au fost reținute și tratate ca pseudo-adnotări, care au fost adăugate iterativ la setul de antrenament. Pentru a îmbunătăți și mai mult performanța, modelul a integrat și raționamentul atât în timpul antrenamentului, cât și al inferenței. Rezultatele au arătat că această strategie a depășit performanța generării few-shot.

Liu et al. (2025) au utilizat un model de dimensiune mică pentru a produce un set mare de exemple de antrenare, care au fost apoi filtrate folosind anumite criterii de calitate (de exemplu, probabilitate, autoconsistență). Doar eșantioanele care au îndeplinit standardele au fost păstrate pentru a forma un set de antrenare filtrat. Modelul de bază a fost reantrenat pe acest set de date, iar ciclul generare-filtrare-reantrenare a fost repetat până la convergența performanței. Acest cadru iterativ a obținut rezultate competitive în task-uri precum parafrazare și rezumare, în ciuda faptului că s-a bazat pe un model de generare mai slab.

T. Wang et al. (2024) și Pang et al. (2024) au avut ca scop îmbunătățirea raționamentului matematic și au introdus un plan de antrenament care încorporează raționamente de tip Chain-of-Thought (CoT). Modelul a fost inițial antrenat pe seturi de date adnotate cu CoT. Ulterior, modelul a fost utilizat pentru a genera pași intermediari de raționament pentru seturi de date care conțineau doar perechi (întrebare, răspuns). Aceste raționamente generate au fost filtrate în exemple bune (care produc răspunsurile corecte)

și exemple slabe (care nu produc răspunsuri corecte) și utilizate pentru a antrena modelul cu DPO (Rafailov et al., 2023).

3 Metodă

3.1 Set de date

Setul de date utilizat în experimentele noastre este FairytaleQA (Xu et al., 2022), un set de date de referință în domeniul educațional. FairytaleQA cuprinde o colecție de povești pentru copii, fiecare însoțită de întrebări și răspunsuri selectate și adnotate de experți în educație. Aceste adnotări sunt deosebit de relevante pentru studiul nostru, deoarece fiecare întrebare este etichetată pe două dimensiuni distincte ale atributelor, și anume, *Focus* și *Coverage*.

Atributul *Focus* indică ținta întrebării. Categoriile tipice includ personajul, acțiunea, cadrul și conflictul, printre altele. Această clasificare ajută la determinarea aspectului poveștii pe care ar trebui să se concentreze un model atunci când generează sau răspunde la o întrebare. În schimb, atributul *Coverage* surprinde nivelul de abstractizare sau domeniul de aplicare textual necesar pentru a răspunde la întrebare. Mai exact, face distincția între întrebările la care se poate răspunde pe baza unei fraze sau propoziții locale, față de cele care necesită integrarea informațiilor în mai multe părți ale poveștii sau rezumarea narațiunii în ansamblu.

Am selectat FairytaleQA datorită utilizării sale largi ca referință de încredere în comunitatea educațională. Construcția sa asigură adnotări de înaltă calitate, aliniate cu obiectivele pedagogice. Este important de menționat că, din câte știm, FairytaleQA este unul dintre puținele seturi de date de generare a întrebărilor disponibile publicului care oferă adnotări de-a lungul mai multor dimensiuni ale atributelor, ceea ce este esențial pentru task-ul nostru de generare și evaluare condiționată de atribute.

Autorii au pre-partiționat setul de date în subseturi de antrenare, validare și testare, constând din 8548, 1025 și respectiv 1007 eșantioane. Cu toate acestea, o analiză detaliată a distribuției valorilor atributelor relevă un grad ridicat de dezechilibru pentru ambele categorii *Focus* și *Coverage* în partiția de antrenament (a se vedea Tabelul 1). Acest dezechilibru prezintă provocări pentru generarea controlată și evaluarea corectă, motivând direct utilizarea tehnicilor de generare sintetică de date și de echilibrare a atributelor, așa cum se discută în secțiunile ulterioare.

3.2 Paradigma de antrenare ORSO

Task-ul în cauză implică utilizarea unui model pentru a genera o pereche întrebare-răspuns pe baza unei intrări structurate. Această intrare constă în: o descriere a sarcinii, un text și un set de valori ale atributelor, cum

Focus	**%**	**Coverage**	**%**
Action	32%	Local	91%
Causal Relation	28%	Summary	9%
Character	11%		
Feeling	10%		
Outcome Resolution	9%		
Setting	6%		
Prediction	4%		

Tabelul 1. Distribuția celor două atribute din FairytaleQA (Focus & Coverage) în partiția de antrenare.

ar fi *Focus* și *Coverage*. Aceste componente sunt combinate într-un singur prompt furnizat LLM-ului. Având o intrare $(Ctx, Attr\_Vals)$, o concatenare a textului și a valorilor atributelor $Attr\_Vals = [attr\_val_1, \ldots, attr\_val_m]$, obiectivul este de a genera o ieșire (Q, A), întrebarea și răspunsul corespunzător.

Totuși, în cazul antrenării supervizate clasice, există o conexiune slabă între diferitele atribute și răspunsul generat. Modelul generează întrebări și răspunsuri fără a le adapta la diferitele configurații ale atributelor și nu învață semantica valorilor acestora. În antrenarea supervizată clasică, maximizăm probabilitatea $P_\theta(Q, A \mid (Ctx, Attr\_Vals)$, dar nu există nicio penalizare dacă modelul generează un rezultat similar pentru o configurație diferită a atributelor $Attr\_Vals'$. Drept urmare, modelul poate ignora condiționarea atributelor și poate ajunge ca $P_\theta(Q, A \mid Ctx, Attr\_Vals')$ să fie aproape de $P_\theta(Q, A \mid (Ctx, Attr\_Vals)$. Modelul nu este penalizat explicit pentru nerespectarea valorilor atributelor, atâta timp cât contextul generat respectă textul și prompt-ul furnizat.

Progresele recente în alinierea LLM-urilor au introdus metode care optimizează generarea moidelului în mod comparativ (de obicei, având aceeași intrare, un model este antrenat să prefere o ieșire dorită, *chosen*, în locul uneia nedorite, *rejected*). Această paradigmă de antrenare, adoptată de Ouyang et al. (2022), Rafailov et al. (2023) sau Hong et al. (2024), este eficientă, dar necesită construirea sau selecția de eșantioane negative, ceea ce este adesea non-trivial, având în vedere spațiul de posibilități mare și divers.

În abordarea noastră, adaptăm acest cadru de aliniere prin introducerea perturbațiilor la nivel de intrare în loc de eșantionarea dintr-un spațiu mare de negative la ieșire. Mai exact, păstrăm ieșirea y fixă și modificăm valorile atributelor în promptul de intrare, creând două variante de intrare: x_c: intrarea cu valori corecte ale atributelor și x_r: intrarea cu valori ale atributelor incorecte sau nepotrivite (eșantionate din domeniul atributelor și în mod deliberat inconsistente cu y). Obiectivul este de a încuraja mo-

delul să acorde atenție variațiilor atributelor și să penalizeze inconsistențele dintre atributele de intrare și ieșirea generată. Acest lucru are ca rezultat o sensibilitate mai bună a atributelor în timpul generării.

Metoda noastră este inspirată de cadrul ORPO (Hong et al., 2024), care nu necesită un model de referință și a obținut performanțe empirice superioare față de DPO și RLHF. Obiectivul de antrenament ORPO este definit ca fiind:

$$\mathcal{L}_{ORPO}(\theta) = \mathcal{L}_{SFT}(\theta) + \lambda \cdot \mathcal{L}_{OR}(\theta) \tag{1}$$

$$\mathcal{L}_{OR}(\theta) = -\log \sigma \left(\log \frac{odds_\theta(y_c \mid x)}{odds_\theta(y_r \mid x)} \right) \tag{2}$$

unde y_c este ieșirea aleasă (chosen), y_r este ieșirea respinsă (rejected), x este intrarea și $odds_\theta(y \mid x) = \frac{P_\theta(y|x)}{1-P_\theta(y|x)}$.

În varianta propusă de noi, ORSO, redefinim loss-ul bazat pe raportul de șanse (vezi Ecuația 2) pentru a opera peste perechi de intrări perturbate în loc de perechi de ieșiri:

$$\mathcal{L}_{OR}(\theta) = -\log \sigma \left(\log \frac{odds_\theta(y \mid x_c)}{odds_\theta(y \mid x_r)} \right) \tag{3}$$

unde x_c este secvența de intrare care conține valorile corecte ale atributelor, x_r este intrarea cu valori ale atributelor în mod deliberat incorecte, iar y este ieșirea de referință corespunzătoare lui x_c.

O comparație vizuală care ilustrează diferența dintre antrenarea supervizată clasică și metoda noastră ORSO este prezentată în Figura 1.

	Prompt			Output
SFT	Context	Focus	Coverage	Question & Answer
	Once upon a time there was a king...	**character**	**local**	Q: What type of ruler was the king? A: kind and just

	Prompt			Output
ORSO	Context	Focus	Coverage	Question & Answer
Chosen	Once upon a time there was a king...	**character**	**local**	Q: What type of ruler was the king? A: kind and just
Rejected	Once upon a time there was a king...	**setting**	**summary**	Q: What type of ruler was the king? A: kind and just

Figura 1. Exemplu comparativ între SFT și ORSO

3.3 Stagii de antrenare

Urmăm același cadru general descris de Dong et al. (2023) pentru SteerLM. Este o abordare semi-supervizată comună, utilizată și de Jung et al. (2024), Morimura et al. (2024) și Liu et al. (2025). Descriem pașii în următoarele subsecțiuni și oferim o prezentare generală în Figura 2.

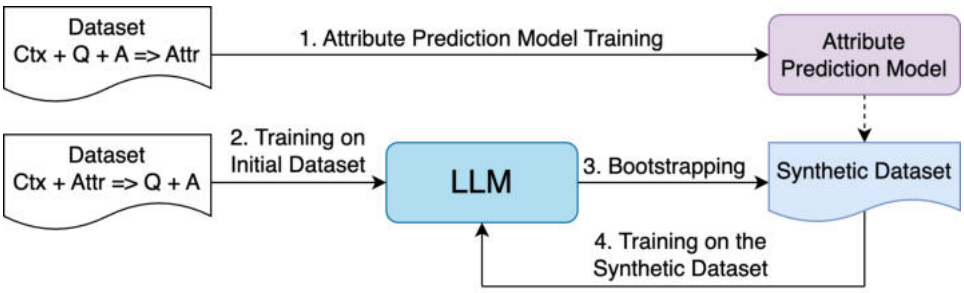


Figura 2. Stagiile generale de antrenare.

3.3.1 Model de predicție a atributelor

Această componentă constă dintr-un set de clasificatori, fiecare antrenat să prezică un atribut specific al întrebării, răspunsului sau contextului generat (de exemplu, Focus, Coverage). Scopul modelului de predicție a atributelor (Attribute Prediction Model) este de a verifica dacă rezultatele generate sunt conforme cu valorile atributelor țintă specificate în promptul de intrare. Un clasificator separat este antrenat pentru fiecare atribut din setul de date. Acești clasificatori sunt utilizați ulterior pentru a evalua performanța atât a metodei propuse (ORSO), cât și a abordării de referință (SteerLM), în ceea ce privește respectarea promptului.

3.3.2 Antrenare pe setul de date inițial

În această etapă, un LLM este antrenat pentru a îndeplini sarcina de generare a unei întrebări și a unui răspuns corespunzător, condiționat de un context dat și de valori ale atributelor specificate (de exemplu, Focus sau Coverage). Procesul de antrenare constă în două faze secvențiale: o epocă inițială de antrenare supervizată clasică pentru a adapta modelul la domeniul sarcinii, urmată de o epocă a antrenamentului bazat pe ORSO descris anterior. Formatul promptului este ilustrat în Figura 3. Cu **bold** marcăm textul pe care modelul trebuie să-l genereze.

3.3.3 Boostrapping

Din cauza dezechilibrului dintre clase în valorile atributelor și a numărului limitat de instanțe de antrenament în general, introducem o etapă de aug-

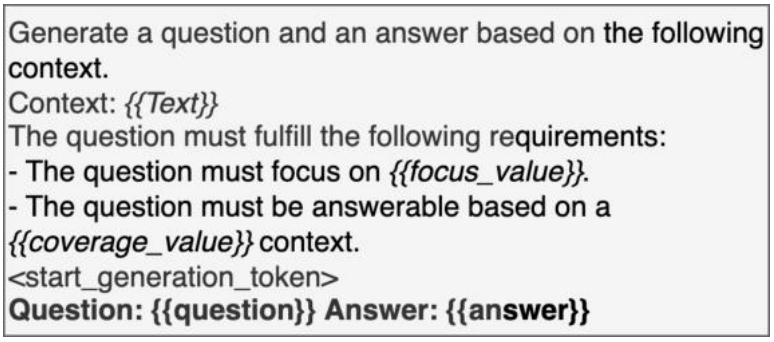


Figura 3. Formatul promptului.

mentare sintetică a datelor.

Mai exact, pentru fiecare text din setul de antrenare original, generăm date sintetice prin eșantionarea ieșirilor ca noi perechi întrebare-răspuns corespunzătoare tuturor combinațiilor posibile de valori ale atributelor. În cazul nostru, generăm 14 noi prompturi per text de intrare (adică, Focus cu 7 valori posibile x Coverage cu 2 valori). Aceste eșantioane generate extind setul de antrenament și sunt utilizate pentru a îmbunătăți generalizabilitatea modelului în toate configurațiile atributelor.

Pentru a îmbunătăți eficacitatea datelor de antrenare sintetic, aplicăm o etapă de filtrare cu 4 obiective principale, derivate din multiple lucrări despre învățare semi-supervizată (Ouyang et al., 2022; Rafailov et al., 2023; Hong et al., 2024): *balansarea numărului de atribute* (un număr egal de exemple pentru fiecare combinație de atribute din setul de date sintetic), *încrederea în generare* (păstrarea doar a acelor eșantioane pentru care modelul prezintă o încredere ridicată în timpul generării), *consistența cu atributele din prompt* (utilizarea modelului de predicție a atributelor antrenat anterior pentru a deduce atributele reale reflectate în fiecare ieșire generată) și *diversitatea exemplelor* (evitarea eșantioanelor excesiv de similare și reducerea redundanței în setul de antrenare).

Distribuția eșantionului rezultată între configurațiile atributelor poate fi în continuare dezechilibrată. Pentru a construi un set de date sintetic echilibrat, ne propunem să păstrăm primele K eșantioane per configurație de atribute. Selecția eșantionului este ghidată de încrederea generării, unde filtrăm exemplele cu o probabilitate sub un prag empiric.

Pentru a promova și mai mult diversitatea semantică, aplicăm clusterizarea eșantioanelor filtrate. Pentru fiecare configurație de atribute, efectuăm clusterizarea de tip k-means, folosind reprezentări vectoriale ale întrebărilor și răspunsurilor obținute printr-un model de encodare (W. Wang et al., 2020). Din fiecare cluster, reținem eșantionul cu cea mai mare probabilitate de generare, asigurând atât diversitatea, cât și calitatea.

Setul de date rezultat cuprinde K mostre sintetice de înaltă calitate și

diverse pentru fiecare configurație de atribute. Acest proces este executat independent pentru SteerLM și ORSO, rezultând două seturi de date sintetice distincte, fiecare cu propriile exemple de antrenare generate folosind modelul respectiv.

3.3.4 Antrenare pe setul de date sintetic

Odată ce setul de date sintetice filtrate este obținut, vom continua cu o fază suplimentară de antrenare pentru a îmbunătăți și mai mult performanța modelului. Spre deosebire de etapa anterioară (a se vedea Secțiunea 3.3.2), această fază constă într-o singură epocă de antrenare bazat pe ORSO. Deoarece modelul a fost deja expus sarcinii prin antrenare supervizată clasică, nu este necesară o epocă suplimentară în această etapă.

3.3.5 Filtrarea configurațiilor de atribute invalide

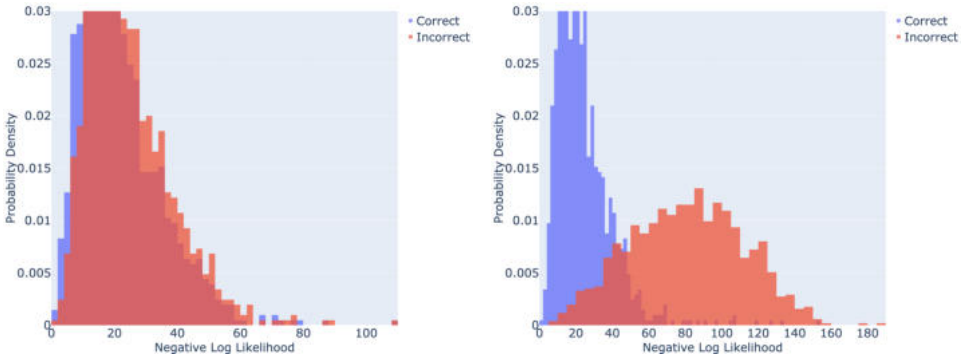
În practică, există contexte de intrare și configurații de atribute pentru care nu se poate genera nicio întrebare validă. O modalitate de a gestiona acest lucru este de a ne baza pe adnotările experților: generăm întrebări doar pentru textele și configurațiile de atribute care au fost confirmate de experți ca fiind valide și în concordanță cu textul. Acest lucru se poate face utilizând atributele adnotate din setul de date FairytaleQA, restricționând generarea la combinațiile text-atribut care apar în adnotări, unde posibilitatea de a genera o întrebare validă este deja stabilită.

Cu toate acestea, este necesar să se includă un mecanism pentru a detecta și comunica această limitare utilizatorului dacă nu sunt disponibile aserțiuni ale experților. Mai exact, modelul ar trebui să evalueze dacă o generare solicitată, condiționată de un anumit context și set de valori ale atributelor, este fezabilă. O abordare este stabilirea unui prag de încredere în generare, bazat pe probabilitatea rezultatului. Dacă încrederea modelului scade sub acest prag, acest lucru ar indica faptul că este probabil ca combinația de atribute solicitată să fie incompatibilă cu contextul de intrare. În astfel de cazuri, sistemul ar trebui să se abțină de la producerea unui rezultat. Acest mecanism ar ajuta la prevenirea producerii de rezultate incoerente sau irelevante semantic de către model, îmbunătățind astfel atât fiabilitatea, cât și interpretabilitatea procesului de generare.

În cazul nostru, pragul este ales folosind setul de validare prin examinarea distribuției valorilor probabilității logaritmice negative pentru exemplele de adnotate de experți și considerând că valorile mai mari de $mean + 1.5 \times std$ sunt potențial nesigure.

3.4 Configurație experimentală

Modelul cu care ne comparăm performanța în acest studiu este inspirat de paradigma de antrenament SteerLM (Dong et al., 2023). Folosim aceiași



(a) SteerLM (b) ORSO

Figura 4. Step 3.3.2: Probabilitatea logaritmică negativă pentru instanțele de atribute corecte versus incorecte.

pași de antrenare descriși în Secțiunea 3.3 atât pentru abordarea noastră, cât și pentru SteerLM. Singura distincție față de SteerLM este că, pentru Pasul 3.3.2 (a doua epocă) și Pasul 3.3.4, antrenarea se face cu abordarea de antrenare supervizată clasică (SFT) pe care SteerLM o propune. În rest, fiecare pas rămâne același, folosind SteerLM în același mod ca abordarea noastră pentru o comparație corectă.

Această configurație controlată ne permite să izolăm contribuția metodei noastre de aliniere, utilizând în același timp SteerLM ca un standard la ora actuală pentru generarea bazată pe atribute.

În experimentele noastre, am utilizat modele de limbaj open-source, alese pentru eficiența lor computațională și pentru adecvarea la medii cu resurse limitate. Ne asigurăm că experimentele noastre rămân accesibile și reproductibile fără a necesita infrastructură costisitoare.

Mai întâi am antrenat doi clasificatori independenți, câte unul pentru fiecare atribut (Focus și Coverage), pentru a implementa modelul de predicție a atributelor (vezi Secțiunea 3.3.1). Am luat în considerare ModernBERT (Warner et al., 2024), un model recent de codificator bazat pe Transformer, și l-am extins cu un cap de clasificare. Fiecare clasificator primește ca intrare o concatenare a contextului, întrebării și răspunsului și prezice valoarea atributului corespunzător. Antrenăm ambii clasificatori pentru o epocă folosind partiția de antrenare a setului de date.

Pentru task-ul de generare a întrebărilor condiționate de atribute, folosim versiunea Instruct a Llama 3.2 1B (Grattafiori et al., 2024) ca model generativ de bază atât pentru abordarea noastră, cât și pentru SteerLM . În timpul fazei de Antrenare pe Setul de Date Inițial (vezi Secțiunea 3.3.2), ambele modele sunt mai întâi antrenate folosind o epocă de învățare supervizată standard (pentru adaptarea la task), urmată de o epocă suplimentară folosind fie SFT în stil SteerLM, fie ORSO (abordarea noastră).

Pentru etapa de bootstrapping, generăm 5 eșantioane pentru fiecare configurație de text și atribute pentru a crea un grup mare de candidați sintetici care să fie selectați pentru antrenament suplimentar. Pe baza analizei din Figura 4, majoritatea eșantioanelor de validare prezintă scoruri de probabilitate logaritmică negativă sub un prag de 100. Acest prag este derivat din probabilitatea modelului de a genera datele de validare selectate de experți și reflectă reprezentările interne ale modelului. În consecință, în timpul bootstrapping-ului, eliminăm toate eșantioanele generate cu o probabilitate logaritmică negativă care depășește 100, deoarece este probabil ca acestea să fie de calitate scăzută. Mai mult, pentru fiecare configurație de atribute, păstrăm $K = 5000$ de eșantioane, rezultând un set de date sintetice de 70k de exemple pentru fiecare model.

3.5 Protocol de evaluare

Inițial, generăm întrebări cu fiecare model, bazate textele din setul de test folosind toate combinațiile posibile de atribute pentru a evalua comportamentul modelului. Ca atare, generăm 5 rezultate candidate pentru fiecare text de intrare și pereche de atribute și o selectăm pe cea cu cea mai mică probabilitate logaritmică negatoivă, presupunând că reprezintă generarea cea mai sigură și coerentă.

O limitare importantă este că unele combinații de atribute pot să nu fie potrivite pentru textul de intrare asociat. De exemplu, o poveste care se concentrează exclusiv pe un personaj poate să nu permită generarea de întrebări semnificative despre decorul scenei. Întrucât evaluarea noastră solicită inițial modelelor să genereze rezultate pentru toate combinațiile de atribute, aceasta include cazuri în care asocierea prompt-atribut este imposibil de realizat.

Pentru a aborda limitările evaluării tuturor combinațiilor de atribute, introducem două noi scenarii de evaluare. În primul rând, luăm în considerare o configurație ideală în care un expert cunoaște posibilele combinații de atribute pentru un anumit text. Acest scenariu este simulat folosind atributele adnotate din partiția de testare, deoarece aceste atribute cuprind un subset al tuturor combinațiilor valide. În al doilea rând, luăm în considerare un scenariu mai realist în care nu avem acces la cunoștințe de specialitate, așa că trebuie să ne bazăm pe o metodă automată pentru detectarea combinațiilor improbabile. Pentru a face acest lucru, ne bazăm pe un prag de probabilitate logaritmică negativă, așa cum este prezentat în Secțiunea 3.3.5.

3.6 Evaluarea performanței

Generarea de întrebări este în mod inerent o sarcină subiectivă, în care pot exista mai multe rezultate valide pentru o anumită intrare. Ca atare, compararea directă între rezultatele generate și întrebările adnotate în setul de

Metric	Focus	Coverage
F1	95.9	99.9
Acc.	96.0	99.9

Tabelul 2. Performanța clasificatorilor.

date nu este o strategie de evaluare fiabilă sau semnificativă. În schimb, având în vedere concentrarea noastră pe generarea controlată pe atribute, evaluăm performanța modelului pe baza a cât de bine se aliniază conținutul generat cu valorile atributelor specificate. O predicție este considerată corectă numai dacă ambele atribute deduse corespund exact cu cele țintă.

3.7 Evaluări umane și bazate pe LLM ale calității întrebărilor generate

Am efectuat un experiment suplimentar pentru a verifica că niciunul dintre modele nu se implică în exploatarea sistemului de recompense, definită ca producerea de rezultate care induc în eroare clasificatorul fără a genera întrebări semantic valide sau de înaltă calitate. Am evaluat calitatea întrebărilor generate (în configurația atributelor adnotate în setul de test) utilizând atât evaluatori umani, cât și LLM-as-a-Judge (GPT-4o). Evaluarea a fost efectuată pe baza a cinci criterii, cu scoruri atribuite între 1 și 5.

4 Rezultate și discuții

Predicția atributelor. Performanța modelelor de predicție a atributelor, deoarece sunt premise în acest studiu, se găsește în Tabelul 2. Aceste valori argumentează că ne putem baza pe acești clasificatori în toate fazele ulterioare.

Evaluările SteerLM versus ORSO. Evaluăm atât SteerLM, cât și ORSO pe setul de date inițial și după bootstrapping (adică, pe setul de date sintetic). Modelele au fost evaluate în trei configurații diferite de atribute: toate combinațiile posibile, doar cele prezente în setul de date și pe baza pragului de probabilitate logaritmică negativă descris în Secțiunea 3.3.5. Toate rezultatele sunt incluse în Tabelul 3.

Rezultatele empirice inițiale au indicat faptul că ORSO gestionează mai bine interacțiunile complexe ale atributelor. Analiza probabilităților modelului a relevat că ORSO a învățat să distingă între solicitările de atribute valide și invalide, în timp ce SteerLM a generat ieșiri aproape agnostic. ORSO, care a fost antrenat explicit să distingă între configurațiile atributelor, nu generează ieșiri cu o probabilitate mare atunci când sunt cerute

atribute incompatibile. Acest lucru este de așteptat, deoarece a fost penalizat în timpul antrenării pentru neconcordanță cu atributele. În schimb, SteerLM tinde să genereze ieșiri indiferent de relevanța atributelor, deoarece îi lipsește un mecanism explicit pentru a impune consecvența atribut-ieșire.

Acest comportament este ilustrat în Figura 4, care arată probabilitatea logaritmică negativă a modelului pentru generarea de perechi întrebări-răspuns selectate de om (din partiția de validare) atât sub prompt-uri de atribute corecte, cât și incorecte. Pentru ORSO, modelul atribuie o probabilitate semnificativ mai mare generărilor sub atributele corecte (albastru) comparativ cu cele incorecte (roșu), indicând o sensibilitate puternică la atribute. SteerLM, pe de altă parte, afișează o distincție minimă, sugerând că are dificultăți în a diferenția între configurațiile atributelor.

Stagiu	**Configurația de atribute**	**F1 (%)**	
		SteerLM	**ORSO**
Inițial - Pasul 3.3.2	Toate combinațiile	41.6	**45.9**
	Adnotate de experți	95.0	**98.6**
Bootstrapping - Pasul 3.3.3	Toate combinațiile	43.9	**56.9**
	Adnotate de experți	97.4	**99.9**
	Filtrate pe bază de prag	44.9	**92.0**

Tabelul 3. Comparația metodelor pentru diferite configurații ale atributelor

Figura 4 și Tabelul 3 indică faptul că modelul nostru depășește performanța SteerLM. Atunci când i se solicită combinații de atribute selectate de experți (plauzibile), acesta se apropie de o precizie aproape perfectă. În plus, menține performanțe robuste în toate configurațiile posibile ale atributelor. Această îmbunătățire provine din capacitatea sporită a modelului de a acorda atenție semanticii atributelor și de a impune alinierea între atributele prompturilor și conținutul generat. În schimb, SteerLM nu reușește să distingă în mod constant între perechile atribut-ieșire valide și invalide. Mai mult, SteerLM are potențialul de a trece cu vederea un atribut atunci când nu este capabil să le respecte pe ambele.

Performanța modelului nostru, așa cum se arată în Tabelul 3, rămâne robustă sub filtrarea automată a combinațiilor de atribute invalide și continuă să depășească SteerLM, indicând faptul că această abordare oferă o aproximare eficientă pentru identificarea generărilor improbabile fără a necesita supraveghere umană în momentul inferenței. Acest lucru contribuie la dezvoltarea unui flux de generare mai fiabil. Mecanismul de prag propus se dovedește a fi extrem de eficient pentru modelul nostru, dar are o utilitate limitată atunci când este aplicat la SteerLM.

Diferența de performanță este și mai vizibilă în Figura 5, care prezintă performanța celor două modele în raport cu numărul de exemple păstrate.

ORSO menține o performanță foarte ridicată, păstrând în același timp jumătate din numărul total de exemple, dovedind performanța robustă a modelului și faptul că alegerea valorii prag nu este critică.

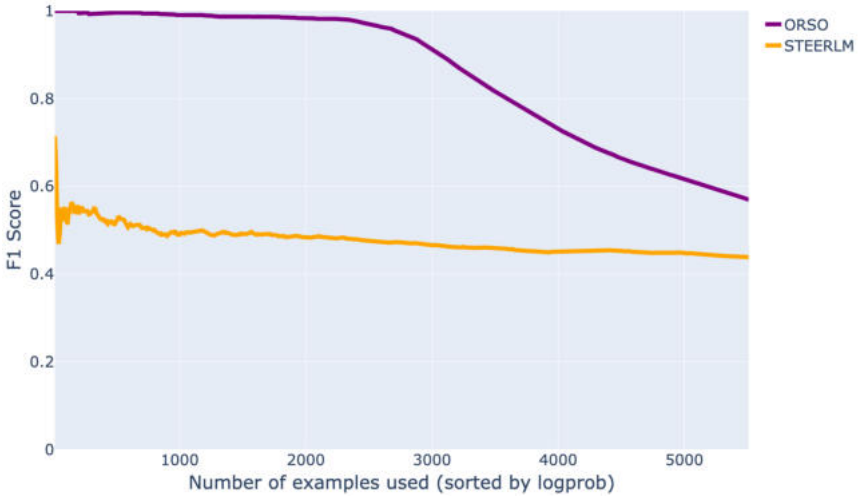


Figura 5. Performanța modelelor pentru diferite praguri de filtrare.

Evaluarea calității întrebărilor. Evaluarea umană a relevat doar o mică diferență în calitatea întrebărilor generate de cele două metode, cu o calitate generală medie de 4,34 pentru ORSO, comparativ cu 4,24 pentru SteerLM . O potențială sursă de eroare apare în evaluările LLM-as-a-Judge, care au favorizat SteerLM (4,52 față de 4,31 în ceea ce privește calitatea generală), în ciuda faptului că adnotările umane au arătat rezultate similare pentru cele două metode. Mai mult, este important de menționat că modelele diferă substanțial în ceea ce privește obiectivele lor de antrenament: ORSO nu a fost optimizat explicit pentru maximizarea calității percepute a rezultatului, ci mai degrabă pentru asigurarea unei conformități mai puternice a atributelor. Această distincție poate explica parțial discrepanța dintre evaluările automate și cele umane, reiterând necesitatea de precauție atunci când ne bazăm pe LLM-as-a-Judge. Cu toate acestea, valorile indicatorilor de calitate rezultați sunt constant ridicate, reflectând cu acuratețe atributele de control.

5 Concluzii și activități viitoare

Acest studiu introduce o metodă inovatoare de antrenare a modelelor pentru generarea de întrebări controlate prin atribute, utilizând o combinație de date selectate de oameni și date sintetice. Abordarea noastră ORSO a

depășit sistematic un standard puternic - SteerLM, atunci când a generat perechi întrebare-răspuns care aderă la atribute specificate (de exemplu, Focus și Coverage). Protocolul nostru experimental a evaluat atât combinații exhaustive de atribute, cât și configurații plauzibile, validate de experți, pentru a identifica diferențele în sensibilitatea atributelor și fidelitatea generării. ORSO a demonstrat în mod constant o sensibilitate ridicată la atribute și o robustețe la variația prompt-ului, validând eficacitatea mecanismului propus. Aceste constatări sugerează că antrenarea explicită bazată pe atribute poate îmbunătăți semnificativ modelele de generare controlabilă.

Pentru lucrările viitoare, intenționăm să explorăm dacă metoda noastră poate fi extinsă eficient la situații în care adnotările atributelor sunt generate sintetic, reducând astfel dependența de etichetele costisitoare ale setului de date, selectate de oameni. Această direcție ar putea permite un control scalabil și adaptabil la domeniu în modelele de generare a întrebărilor, menținând în același timp alinierea cu atributele de conținut specificate.

Limitări

Cadrul ORSO propus îmbunătățește semnificativ gradul de conștientizare a atributelor în generarea întrebărilor; cu toate acestea, anumite aspecte pot afecta aplicabilitatea sa mai largă.

ORSO, precum și SteerLM, depind de disponibilitatea unui clasificator de atribute precis. Acest clasificator joacă un rol central atât în evaluare, cât și în filtrarea generării. În situațiile în care un astfel de clasificator nu este suficient de precis sau nu este disponibil, semnalele de control utilizate pentru a evalua sau ghida generarea pot fi mai puțin fiabile. Deși aceasta este o cerință generală pentru abordările actuale în generarea controlată prin atribute, este important de remarcat faptul că lucrările viitoare ar putea beneficia de explorarea abordărilor care reduc dependența de clasificatorii externi.

Mai mult, modelele generative sunt în mod inerent înclinate să producă rezultate chiar și în condiții de combinații de atribute invalide sau contradictorii. Pentru a rezolva această problemă, am introdus un mecanism de prag bazat pe încrederea în generare, suprimând rezultatele care este puțin probabil să îndeplinească constrângerile date. Deși această soluție s-a dovedit adecvată pe setul de date utilizat în experimentele noastre, pragul s-ar putea să nu se transfere direct în alte instanțe. Sunt necesare experimente suplimentare pentru a studia posibilitatea unor strategii generalizabile sau a unor clasificatori de fezabilitate.

Acești factori nu limitează contribuțiile principale ale ORSO, ci sugerează direcții pentru îmbunătățirea robusteții și a pregătirii pentru implementare în medii mai diverse.

Bibliografie

Dong, Y., Wang, Z., Sreedhar, M., Wu, X., & Kuchaiev, O. (2023). Steerlm: Attribute conditioned sft as an (user-steerable) alternative to rlhf. In *Findings of the association for computational linguistics: Emnlp 2023* (pp. 11275–11288).

Gheshlaghi Azar, M., Daniel Guo, Z., Piot, B., Munos, R., Rowland, M., Valko, M., & Calandriello, D. (2024, 02–04 May). A general theoretical paradigm to understand learning from human preferences. In S. Dasgupta, S. Mandt, & Y. Li (Eds.), *Proceedings of the 27th international conference on artificial intelligence and statistics* (Vol. 238, pp. 4447–4455). PMLR. Retrieved from `https://proceedings.mlr.press/v238/gheshlaghi-azar24a.html`

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., ... others (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Guo, Y., Cui, G., Yuan, L., Ding, N., Sun, Z., Sun, B., ... others (2024). Controllable preference optimization: Toward controllable multi-objective alignment. In *Proceedings of the 2024 conference on empirical methods in natural language processing* (pp. 1437–1454).

Hong, J., Lee, N., & Thorne, J. (2024). Orpo: Monolithic preference optimization without reference model. In *Proceedings of the 2024 conference on empirical methods in natural language processing* (pp. 11170–11189).

Jung, J., West, P., Jiang, L., Brahman, F., Lu, X., Fisher, J., ... Choi, Y. (2024, June). Impossible distillation for paraphrasing and summarization: How to make high-quality lemonade out of small, low-quality model. In K. Duh, H. Gomez, & S. Bethard (Eds.), *Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 1: Long papers)* (pp. 4439–4454). Mexico City, Mexico: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2024.naacl-long.250/` doi: 10.18653/v1/2024.naacl-long.250

Li, K., & Zhang, Y. (2024). Planning first, question second: An llm-guided method for controllable question generation. In *Findings of the association for computational linguistics acl 2024* (pp. 4715–4729).

Liu, C., Chao, Q., Zhang, W., Wu, X., Li, B., Tuan, L. A., & Bing, L. (2025). Zero-to-strong generalization: Eliciting strong capabilities of large language models iteratively without gold labels. In *Proceedings of the 31st international conference on computational linguistics* (pp. 3716–3731).

Morimura, T., Sakamoto, M., Jinnai, Y., Abe, K., & Ariu, K. (2024). Filtered direct preference optimization. In *Proceedings of the 2024 conference on empirical methods in natural language processing* (pp. 22729–22770).

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... others (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, *35*, 27730–27744.

Pang, R. Y., Yuan, W., He, H., Cho, K., Sukhbaatar, S., & Weston, J. (2024). Iterative reasoning preference optimization. *Advances in Neural Information Processing Systems*, *37*, 116617–116637.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., & Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, *36*, 53728–53741.

Tu, L., Yavuz, S., Qu, J., Xu, J., Meng, R., Xiong, C., & Zhou, Y. (2024). Unlocking anticipatory text generation: A constrained approach for large language models decoding. In *Proceedings of the 2024 conference on empirical methods in natural language processing* (pp. 15532–15548).

Wang, T., Li, S., & Lu, W. (2024). Self-training with direct preference optimization improves chain-of-thought reasoning. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 11917–11928).

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., & Zhou, M. (2020). Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, *33*, 5776–5788.

Warner, B., Chaffin, A., Clavié, B., Weller, O., Hallström, O., Taghadouini, S., ... others (2024). Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv:2412.13663*.

Xu, Y., Wang, D., Yu, M., Ritchie, D., Yao, B., Wu, T., ... others (2022). Fantastic questions and where to find them: Fairytaleqa–an authentic dataset for narrative comprehension. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 447–460).