BABEȘ-BOLYAI UNIVERSITY, CLUJ-NAPOCA, ROMANIA

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Face2Learn

– MIRPR Report 2025 –

Team members:

Bucur Victor Sever, Popoviciu Luca, Porcar Cezar, Potra-Rațiu Darius, Preduca Matei

2025

Rezumat

Aplicația reprezintă un sistem inteligent bazat pe inteligență artificială, denumit **Face2Learn**, care combină modele de limbaj mari (LLM), sinteză vocală (TTS) și animație facială pentru a crea o experiență de învățare naturală și interactivă. Scopul proiectului este de a transforma informațiile academice în explicații conversaționale, vizuale și auditive, crescând astfel implicarea, accesibilitatea și retenția informației. Performanța sistemului este evaluată prin acuratețea răspunsurilor, latența end-to-end și naturalețea percepută a avatarului.

Cuprins

Capitolul 1

Descrierea problemei rezolvate cu ajutorul AI

1.1 Contextul problemei

Sistemele educaționale tradiționale se bazează predominant pe text și prelegeri statice. În contextul actual al digitalizării, apare nevoia de metode de predare interactive, personalizate și accesibile. Proiectul **Face2Learn** propune utilizarea inteligenței artificiale multimodale pentru a crea un asistent virtual care explică concepte academice prin vorbire și expresii faciale sincronizate.

Modelul AI poate înțelege întrebări formulate în limbaj natural, poate accesa informații relevante din materiale educaționale și poate furniza răspunsuri clare, exprimate vocal și vizual printr-un avatar animat.

1.2 Scopul și importanța problemei

Scopul principal este de a transforma procesul de învățare într-o experiență naturală, captivantă și accesibilă. Importanța proiectului derivă din:

- creșterea **engagement-ului** și motivației elevilor/studenților;
- asigurarea **accesibilității** pentru persoane cu deficiențe de vedere sau auz;
- sprijinirea cadrelor didactice prin automatizarea explicațiilor și a sesiunilor de Q&A;
- posibilitatea **învățării personalizate** în ritmul fiecărui utilizator.

1.3 Utilizatorii sistemului

- **Studenți și elevi** – folosesc avatarul AI pentru explicații și recapitulări;
- **Profesori și tutori** – utilizează sistemul pentru demonstrații și asistență automată;

- **Instituții educaționale** – integrează soluția pentru suport didactic 24/7;
- **Persoane cu dizabilități** – beneficiază de conținut multimodal adaptat.

1.4 Datele de intrare și ieșire

Date de intrare:

- întrebări formulate în limbaj natural (text sau voce);
- documente educaționale (manuale, cursuri, notițe);
- preferințe ale utilizatorului (limbă, voce, tonalitate).

Date de ieșire:

- răspuns text generat de LLM;
- voce sintetică naturală generată prin TTS;
- videoclip animat cu avatar sincronizat cu vorbirea.

1.5 Tipurile de date utilizate

- corpusuri textuale academice și explicații didactice;
- înregistrări audio pentru antrenarea TTS;
- imagini/video cu expresii faciale pentru sincronizare (lip-sync);
- embeddings semantice pentru căutare contextuală (RAG).

1.6 Măsurarea performanței sistemului AI

Performanța sistemului este evaluată prin indicatori cantitativi și calitativi:

- **Acuratețea răspunsurilor** – procentul de răspunsuri corecte;
- **Timpul mediu de răspuns** – durata procesării end-to-end;
- **Calitatea animației** – gradul de sincronizare buze-vorbire;
- **Consum de resurse** – memorie și timp de inferență.

Capitolul 2

Related work and useful tools and technologies

Această secțiune prezintă zece proiecte și tehnologii relevante pentru construcția unui avatar AI educațional. Pentru fiecare sunt menționate tipul datelor folosite, algoritmii utilizați, performanțele și tehnologiile implicate.

2.1 1. LoRA (Low-Rank Adaptation)

Date: text educațional.
Algoritmi: fine-tuning eficient al LLM-urilor prin adaptare low-rank.
Performanță: îmbunătățire a acurateței cu cost redus.
Tehnologii: PyTorch, Hugging Face, GitHub open-source.

2.2 2. QLoRA

Date: corpus text.
Algoritmi: fine-tuning cu cuantizare pentru reducerea memoriei.
Performanță: menține calitatea modelului la 4-bit.
Tehnologii: Transformers, bitsandbytes, Hugging Face.

2.3 3. llama.cpp

Date: text.
Algoritmi: inferență locală pentru modele cuantizate.
Performanță: latență redusă pe CPU.
Tehnologii: C++, GGUF models, GitHub.

2.4 4. TinyLLM (Phi, Mistral, TinyLLaMA)

Date: text.
Algoritmi: modele compacte pentru rulare eficientă.
Performanță: raport bun între viteză și acuratețe.
Tehnologii: PyTorch, Hugging Face.

2.5 5. RAGFlow

Date: documente și note de curs.
Algoritmi: RAG (retrieval augmented generation).
Performanță: răspunsuri mai relevante.
Tehnologii: LangChain, FAISS, GitHub.

2.6 6. RAG-Anything

Date: fișiere locale (PDF, text).
Algoritmi: flux simplificat RAG.
Performanță: acces rapid la surse externe.
Tehnologii: Python, Streamlit, GitHub.

2.7 7. Whisper TTS

Date: corpusuri audio şi transcripţii text.
Algoritmi: model neural de sinteză vocală bazat pe arhitectura Whisper.
Performanţă: voce naturală şi suport multilingv.
Tehnologii: PyTorch, Whisper TTS API (OpenAI), Hugging Face, GitHub.

2.8 8. Piper

Date: audio/text.
Algoritmi: TTS optimizat pentru dispozitive edge.
Performanță: latență foarte mică.
Tehnologii: Rust, on-device inference.

2.9 9. Wav2Lip

Date: video + audio.
Algoritmi: lip-sync bazat pe rețele CNN.

Performanță: aliniere buze-vorbire realistă.
Tehnologii: PyTorch, OpenCV, GitHub.

2.10 10. SadTalker

Date: imagine + audio.
Algoritmi: talking-face generation dintr-o singură imagine.
Performanță: expresii faciale naturale.
Tehnologii: PyTorch, DeepFace, GitHub.

Capitolul 3

Evaluarea sistemului RAG utilizat în Face2Learn

3.1 Descrierea și explorarea datelor (EDA)

Pentru evaluarea componentei de **întrebare-răspuns** din sistemul Face2Learn, a fost creat manual un set de date format din **50 de perechi întrebare–răspuns**. Fiecare întrebare a fost extrasă din conținutul manualului `manual2022.pdf`, iar răspunsul aferent reprezintă transcrierea exactă a fragmentului relevant.

Structura fișierului de date (`evaluare.json`)**:**

```
[
  {
    "intrebare": "Ce reprezintă un segment orientat?",
    "raspuns_asteptat": "un segment ... unde s-a precizat originea si extremitatea"
  },
  ...
]
```

Preprocesare:

- Eliminarea diacriticelor pentru uniformitate.
- Conversia la litere mici și eliminarea spațiilor multiple.
- Nu s-a aplicat tokenizare, deoarece evaluarea folosește similaritate semantică și ROUGE.

Explorare sumară:

- Număr total de exemple: 100;
- Lungime medie întrebare: 9 cuvinte;

- Lungime medie răspuns: 12 cuvinte;
- Domeniu: concepte geometrice (vectori, segmente, egalitate etc.);
- Tip date: text scurt, conceptual – ideal pentru evaluarea unui sistem RAG.

3.2 Descrierea algoritmului inteligent

Algoritm ales: *Retrieval-Augmented Generation (RAG).*

Motivație: Arhitectura RAG combină avantajele modelelor de căutare contextuală (retrieval) cu cele de generare (generation), permițând sistemului să răspundă coerent pe baza unui context relevant extras dintr-o bază de cunoștințe (PDF-ul cursului). Această abordare elimină necesitatea antrenării unui model mare de la zero, reducând costurile și riscul de halucinații.

Componente principale:

- **Retrieval:** Model de embedding `thenlper/gte-small` (din `sentence-transformers`) și indexare cu `faiss-cpu`. Scopul este transformarea fragmentelor PDF în vectori semantici și extragerea celor mai relevante pasaje.
- **Generation:** Modelul `mistral-7b-instruct-v0.3.Q4_K_M.gguf`, rulat local prin LM Studio. Acesta generează răspunsul final folosind contextul returnat de retriever.

Librării utilizate: `langchain`, `sentence-transformers`, `faiss-cpu`, `numpy`, `transformers`, `sklearn.metrics`.

3.3 Metodologia experimentală

Împărțirea datelor: Setul de 50 de exemple a fost folosit în întregime ca **set de test**. Scopul principal a fost evaluarea sistemului complet (RAG + LLM), nu antrenarea unui model nou.

Hiperparametri utilizați:

Parametru	Valoare	Descriere
chunk_size	256	Lungimea unui fragment de text la indexare
chunk_overlap	25	Suprapunerea dintre fragmente consecutive
k	4	Numărul de pasaje returnate de retriever
temperature	0.3	Controlul creativității LLM-ului

Metrici de evaluare:

- **ROUGE-L (F1):** măsoară suprapunerea exactă între răspunsul generat și cel așteptat;
- **Similaritate Semantică (Cosine Similarity):** măsoară apropierea conceptuală dintre răspunsuri, folosind embedding-urile `gte-small`.

3.4 Rezultate obținute

Metrică	Valoare medie
ROUGE-L (F1)	20.61%
Similaritate Semantică	90.24%

Interpretare:

- Scorul semantic ridicat (90%) indică faptul că sistemul a înțeles corect întrebările și a generat răspunsuri cu același sens;
- Scorul ROUGE redus (20%) arată că modelul preferă să reformuleze textul în loc să reproducă exact pasajul original.

Exemple reprezentative:

Caz	Întrebare	Răspuns așteptat	Răspuns generat	Observație
1	Segment orientat	un segment ... cu origine și extremitate	o porțiune ... cu direcție asignată	sens corect, formulare diferită
2	Egalitate segmente	A=C și B=D	A=D și B=C	halucinație (ordine inversată)
3	Vector liber	o clasă de echivalență	o familie de vectori legați	parafrazare semantică

3.5 Analiza finală

Diferența semnificativă dintre scorurile ROUGE și Similaritate Semantică evidențiază o problemă frecventă în sistemele RAG:

- Modelul LLM **înțelege contextul**, dar nu respectă instrucțiunea „răspunde exclusiv pe baza textului oferit";
- Răspunsurile sunt logic corecte, dar lexical diferite;
- În unele cazuri apar **halucinații minore** (exemplul #2).

Concluzie parțială: Rezultatele arată că sistemul RAG implementat are o **performanță semantică excelentă**, dar necesită îmbunătățiri și aplicarea unei penalizări de diversitate în prompt.

Concluzii

Proiectul „Face2Learn” oferă o abordare inovatoare de integrare a AI multimodal (text, voce, video) în domeniul educației. Combinația între LLM-uri optimizate, TTS și animație sincronizată contribuie la îmbunătățirea experienței de învățare, făcând-o mai naturală, mai interactivă și mai accesibilă.