

Romanian Open-Source Chatbot

Candidate:

Andrei – Răzvan, JOLDEA

Scientific coordinators:

Associate Professor Dr. Eng. Cosmin, CERNĂZANU,
PhD Student Eng. Andrei, BULZAN

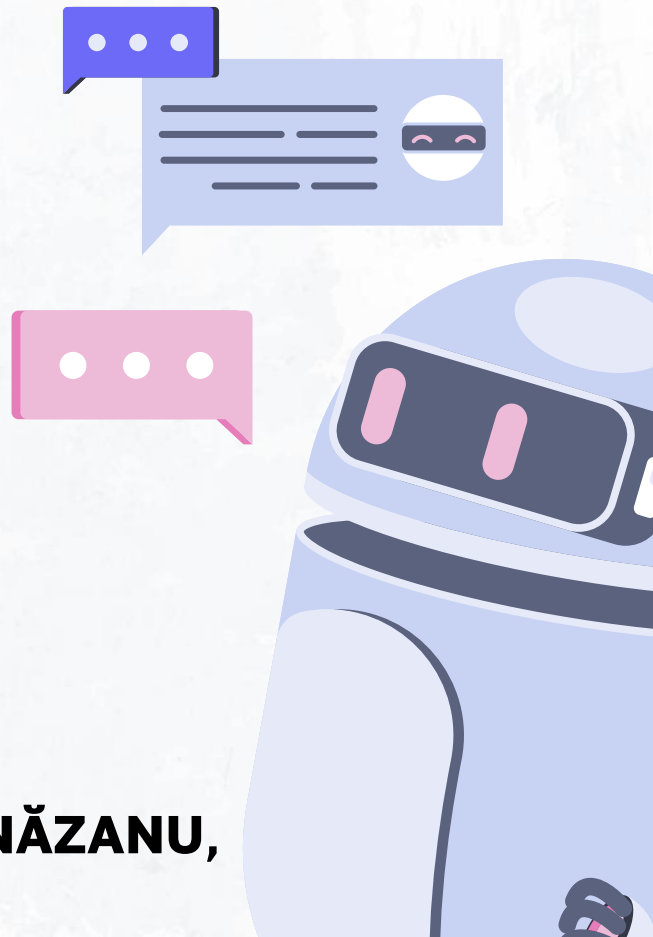


Table of contents

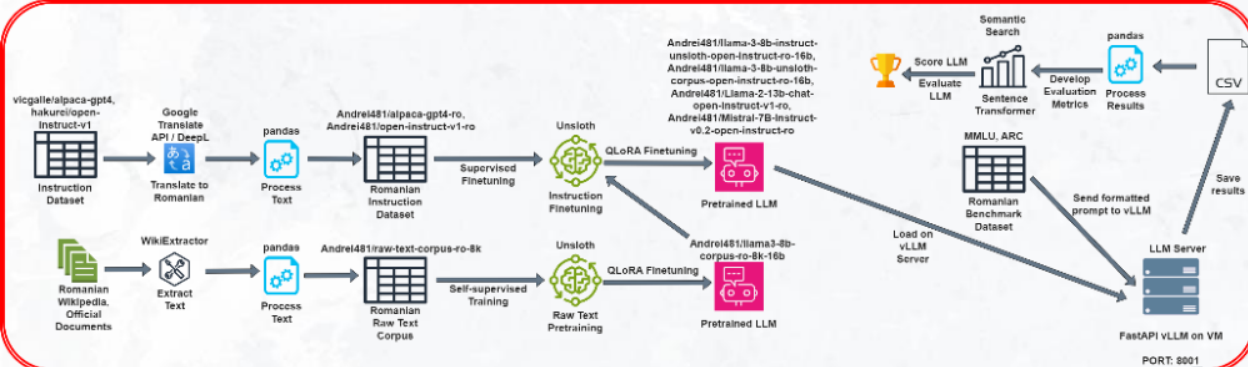
- 01** → **Introduction**
- 02** → **Implementation**
- 03** → **Experimental Results**
- 04** → **Conclusions**

01 →

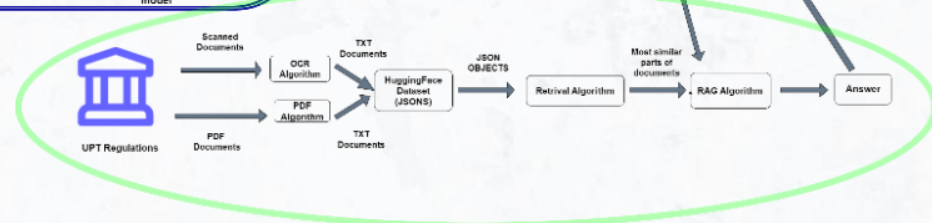
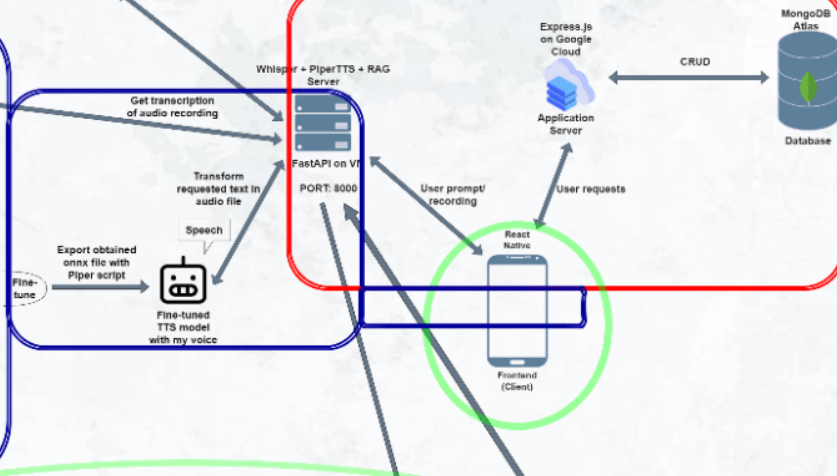
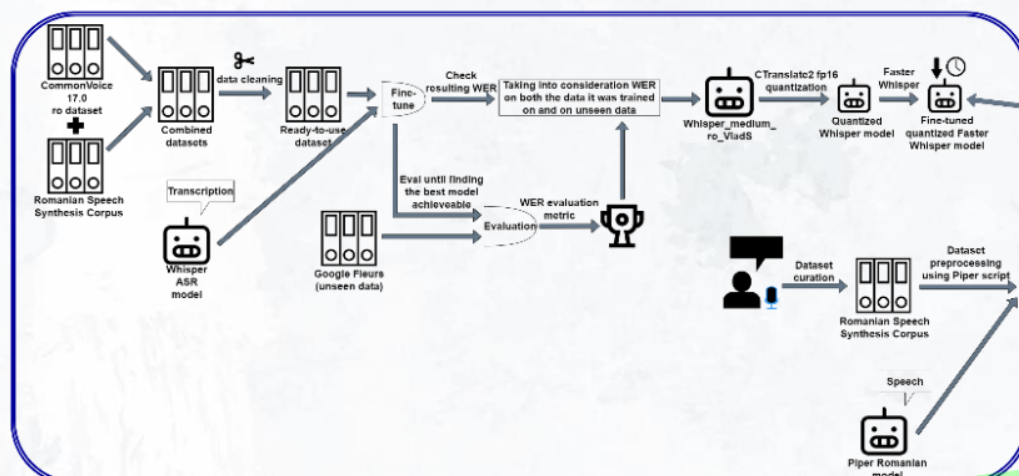
Introduction

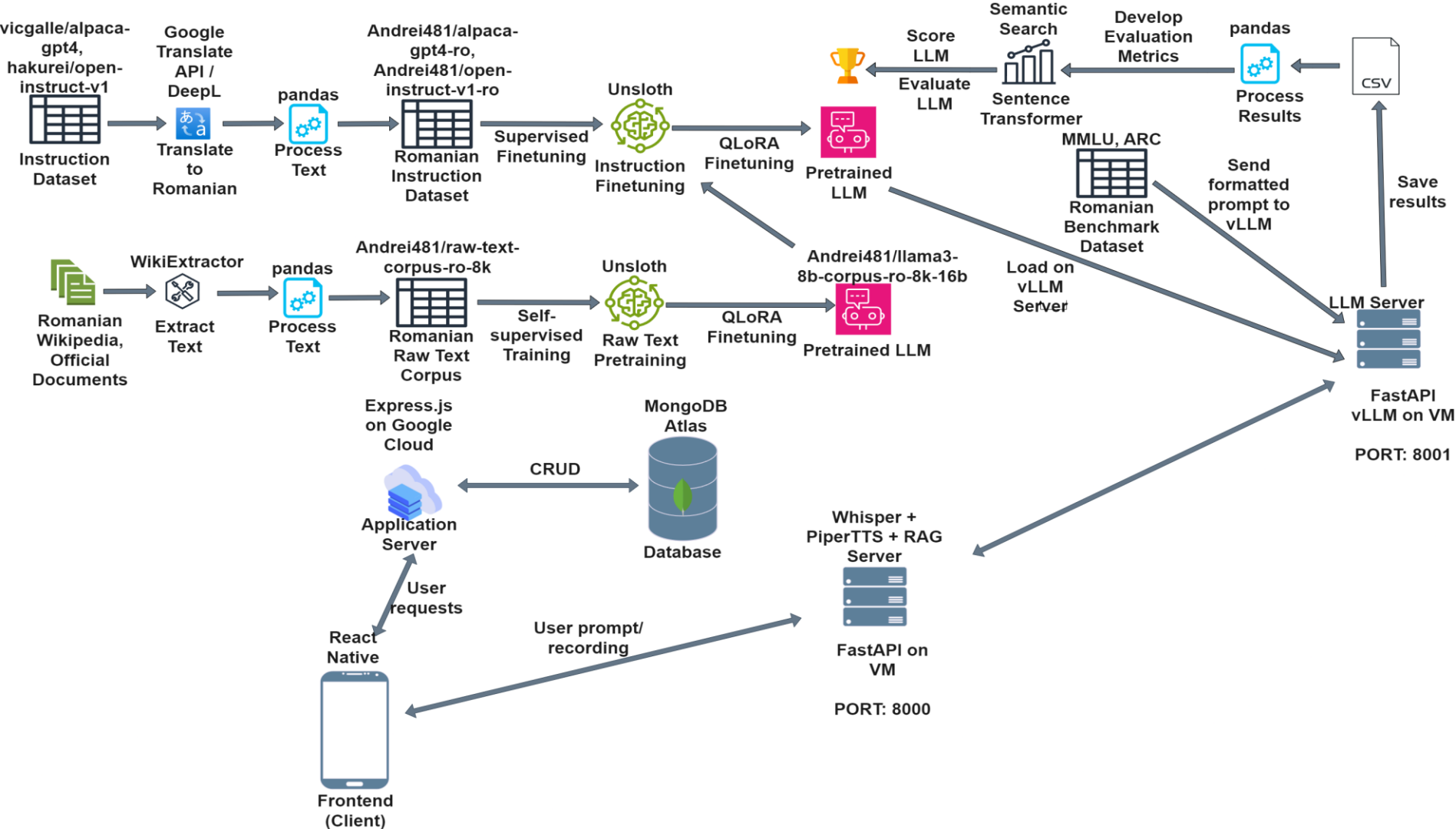
Introduction

- **Domains Covered:**
 - **NLP**
 - **Generative AI**
 - **Mobile Application Development**
- **Part of an ampler project, developed with Diana Cernăzanu and Vlad Sârbu.**
- **Goal:**
 - **Create an open-source LLM capable of responding in Romanian.**
- **This presentation covers solely my contribution.**



- Andrei JOLDEA
- Vlad SÂRBU
- Diana CERNĂZANU





02 →

Implementation

Data Preparation

Finetune dataset:

- **hakurei's open-instruct-v1 ⇒ DeepL ⇒ open-instruct-v1-ro**

Pretraining Dataset:

- **Romanian Wikipedia Dump + Romanian Constitution ⇒ raw-text-corpus-ro-8k:**
 - **553MB Pretraining dataset**

Base Model Finetuning

Base Model	Llama 2 13B Chat (Finetuned)	Llama 3 8B Instruct (Finetuned)	Llama 3 8B (Pretrained + Finetuned)	Mistral 7B Instruct v0.2 (Finetuned)
Dataset	open-instruct-v1-ro	open-instruct-v1-ro	raw-text-corpus-ro-8k + open-instruct-v1-ro	open-instruct-v1-ro
Training Duration	84h	53h	163h + 54h	49h

- **Based on the Transformer architecture**
- **Unsloth** \Rightarrow speed up training
- **QLoRA** \Rightarrow efficient finetuning
- **Finetune** \Rightarrow **3 epochs**
- **Pretraining** \Rightarrow **1 epoch**

LLM Evaluation

Gather Romanian benchmarking datasets:

- **Okapi's Romanian-translated MMLU and ARC datasets**

Develop metrics and score models

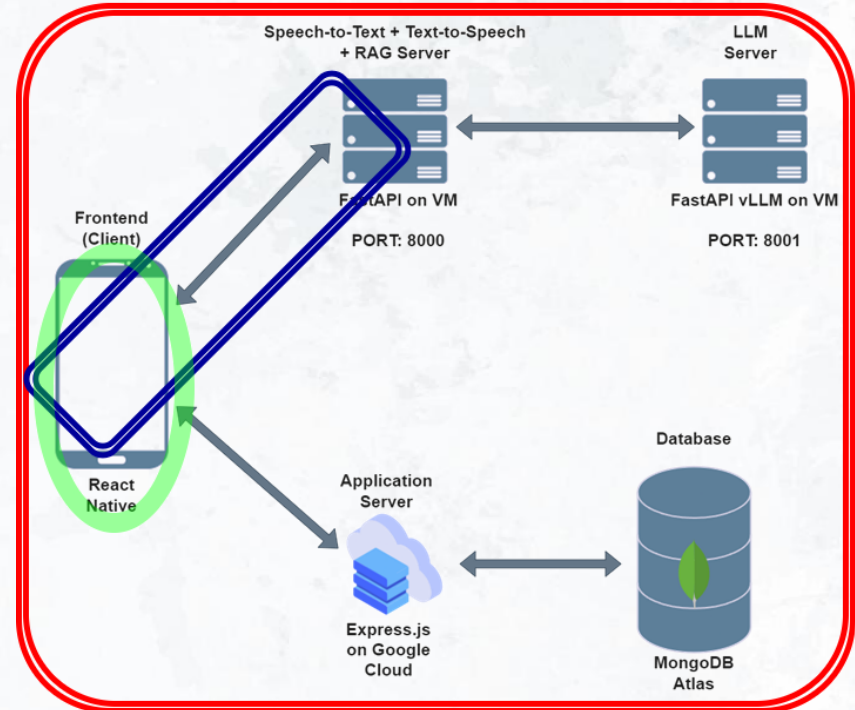
```
"id": "ARC-Challenge/test/Mercury_7271513",  
"answer": "D",  
"instruction": "Care dintre următoarele este o dovadă pentru teoria plăcilor tectonice?",  
"option_a": "topirea calotelor glaciare polare",  
"option_b": "deplasarea biomurilor odată cu schimbările climatice",  
"option_c": "prevalența granitului pe toate continentele",  
"option_d": "prezența vulcanilor din jurul marginii Pacificului"
```

```
"instruction": "Formula lui Drake încercă să răspundă la următoarea întrebare:",  
"option_a": "Va deveni Soarele o gaură neagră?",  
"option_b": "Este universul infinit de mare?",  
"option_c": "Cât de veche este universul vizibil?",  
"option_d": "Suntem singuri în univers?",  
"answer": "D",  
"id": "astronomy/test/103"
```

Mobile Application Development

Architecture:

- MongoDB Atlas Database
- React Native frontend
- Node.js server running in Google Cloud
- Two FastAPI servers:
 - Speech-to-text, text-to-speech, RAG
 - vLLM



03 →

Experimental Results

Experimental Results

Benchmark/Model	Llama3 8B RO (Pretrained + Finetuned)	Llama3 8B RO (Finetuned)	Llama 3 8B	Mistral 7B Instruct v0.2 (Finetuned)	Mistral 7B
ARC (Zero-Shot)	35.22	43.81	32.82	29.64	26.48
MMLU (Zero-Shot)	33.31	42.11	30.34	26.12	25.99
Total Score	34.26	42.96	31.58	27.88	26.23

Llama 3 8B Finetuned ⇒ **36.03%** increase compared to **Llama 3 8B Base**

Llama 3 8B Pretrained + Finetuned ⇒ **8.49%** increase compared to **Llama 3 8B Base**

Mistral 7B Finetuned ⇒ **6.29%** increase compared to **Mistral 7B Base**

Llama 2 13B Finetuned ⇒ improved reasoning and responses in Romanian

Care sunt condițiile necesare pentru a avea un deadlock în programare?[/INST]



Llama2 13B Chat

Un deadlock este o situație în care două sau mai multe procese se află în starea de blocare reciprocă, fiecare proces awaiting for resources that are held by another process, and vice versa. To have a deadlock, the following conditions must be met:

1. There must be two or more processes involved.
2. Each process must be waiting for a resource that is held by another process.
3. Each process must be preventing other processes from accessing the resources they need.

For example, if one process is waiting for a resource that is held by another process, but that other process is waiting for a different resource that is held by the first process, then a deadlock occurs.

It's important to note that deadlocks can be avoided using techniques such as resource allocation, priority scheduling, and resource ordering.

Care sunt condițiile necesare pentru a avea un deadlock în programare?[/INST]



Llama2 13B Chat Romanian

Condițiile necesare pentru a avea un deadlock în programare includ două sau mai multe procese concurente care se găsesc într-o stare de așteptare unul față de celălalt și în același timp amândouă au nevoie ca celelalte procese să completeze anumite sarcini înainte de a putea continua cu propriile activități.

04 →

Conclusions

Conclusions

- Each finetuned model shows improvements over its base model.
- The mobile app enables easier access to UPT-related information through chat with the LLM.
- The base model's language capability can be enhanced with minimal pretraining exposure.
- This can be achieved on consumer-based hardware.

Future Work

- **Improve finetuned models using Reinforcement Learning with Human Feedback.**
- **Experiment with other open-source base models.**
- **Improve the mobile application's functionality.**
- **Create a web application.**

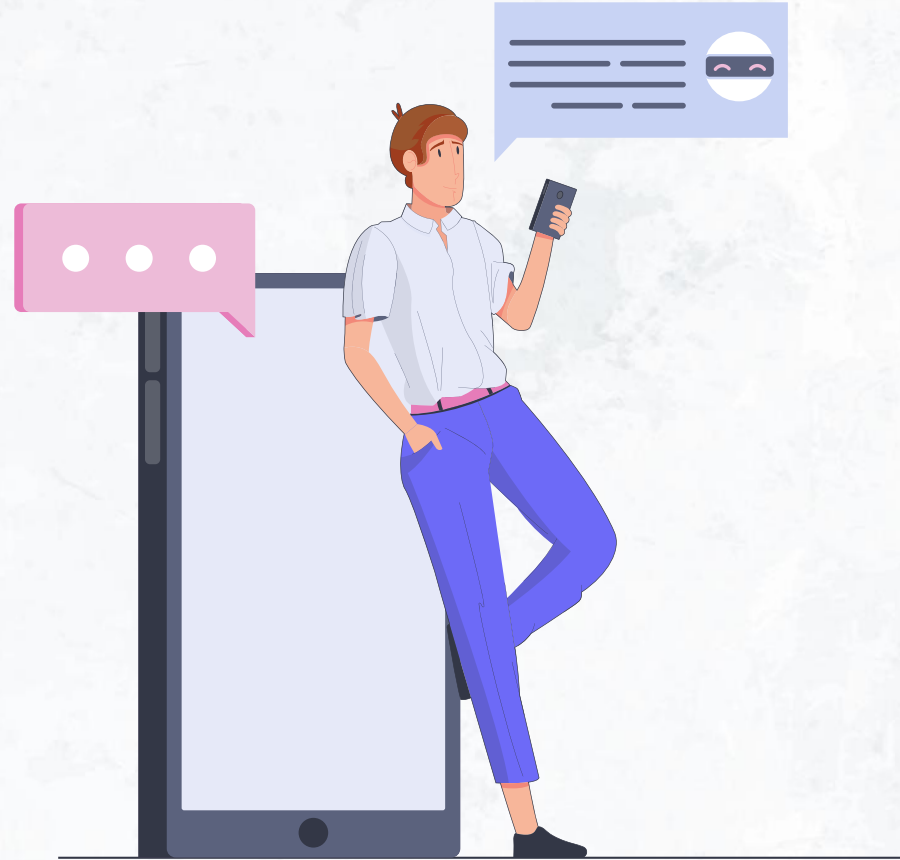
References

1. Lai, Viet Dac, et al. "Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback." arXiv preprint arXiv:2307.16039 (2023).
2. <https://huggingface.co/datasets/hakurei/open-instruct-v1>
3. <https://unsloth.ai/introducing>
4. Dettmers, Tim, et al. "Qlora: Efficient finetuning of quantized llms." Advances in Neural Information Processing Systems 36 (2024).

Thank you!

Any questions?

CREDITS: This presentation template was created by **Slidesgo** and includes icons by **Flaticon**, infographics & images by **Freepik** and content by **Eliana Delacour**



Video

Link:

https://drive.google.com/file/d/13jz8R7PMhtS2uEVnJGck_UcDLFlZApO7/view?usp=sharing