AI Hackathon, Briefing for Dev Team

Llama 3.1

Remerge

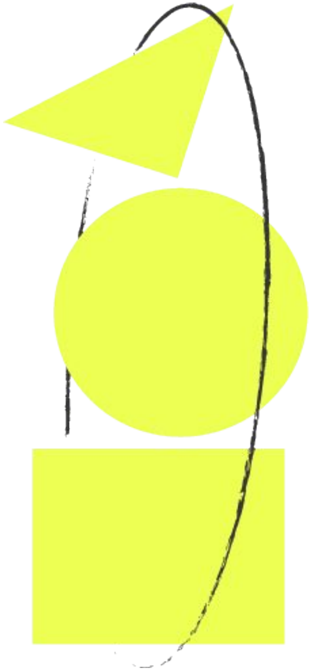
ChatGPT

18.12.2024

AI Productivity Team

Agenda

- Introduction: Event objectives
- Pros and cons, proprietary and open source LLMs
- Examples with OpenAI: simple chat bot => with instructions => with memory => RAG/assistant/Confluence => function calling
- Examples with Llama3: simple chat bot => with instructions => with memory => RAG/assistant/Confluence => image identification
- Feedback and questions



Event objectives:

1. Increase productivity
2. Empower users with AI tools
3. Enable task automation
4. Encourage experimentation and hands-on learning

AI Team:

Alan

[redacted]

[redacted]

You can work on teams or on your own

Presentation’s time is Friday at 15:00 (CET)

What is a LLM?

A Large Language Model (LLM) is an advanced artificial intelligence system trained on vast datasets to understand, process, and generate human-like text. These models, like OpenAI's GPT or Llama, are based on deep learning architectures such as transformers. They excel in a variety of language-related tasks, including:

Text generation
Question answering
Sentiment analysis
Summarization
Translation
Information retrieval

LLMs achieve this by learning patterns, structures, and contextual meanings in language, enabling them to produce coherent and contextually relevant responses.

What make LLMs powerful?

1. **Model**
   - A mathematical representation with **billions of parameters**.
   - Captures language patterns and relationships.
2. **Temperature**
   - Controls **randomness** in responses:
     - **Low (0.1):** Accurate and repetitive.
     - **High (0.8):** Creative and diverse.
3. **Max Tokens**
   - Sets the **maximum length** of the response.
   - Tokens = words or word fragments.
4. **Prompt & Context Window**
   - **Prompt**: The instruction given to the model.
   - **Context Window**: Maximum tokens processed (GPT-4 up to 32K).

5. **Top-p (Nucleus Sampling)**

   - Controls **cumulative probability**:
     - **Low (0.3):** Most likely options.
     - **High (0.9):** More diverse choices.

6. **Stop Sequences**

   - Defines words/phrases where **text generation stops**.

7. **Pre-training & Fine-tuning**

   - **Pre-training**: General training on massive data.
   - **Fine-tuning**: Adjustment for specific tasks.

Pros and cons proprietary and open source LLMs

Pros:

- State-of-the-Art Models
- Ease of Use
- Versatility
- Continuous Improvement
- Cloud-Based
- Scalability

Cons:

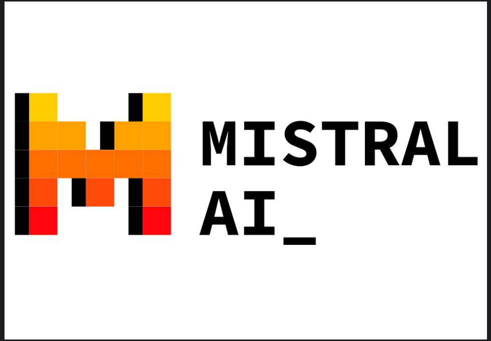
- Security Concerns
- Cost
- Dependence on External Service
- Latency
- Limited Customization

Pros:

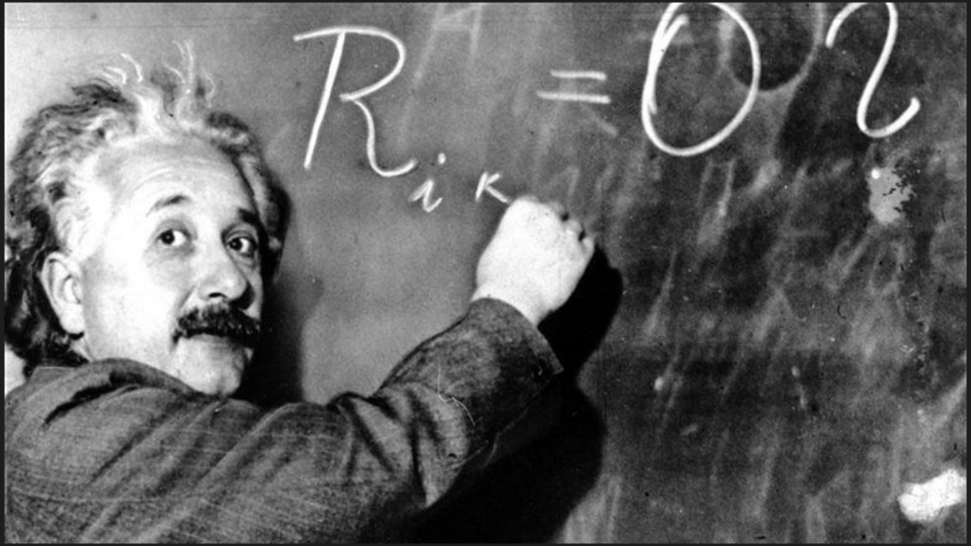
- Cost-Effective
- Customizability
- Data Privacy
- Diverse Use Cases
- Community Support
- Flexibility

Cons:

- Computational Resources
- Maintenance Overhead
- Steeper Learning Curve
- Performance Variability
- Limited Support



$E = mc^2$… or maybe it’s just time to get started.



We will be using the following OpenAI Interfaces:

Chat Completion API: For quick text generation tasks where you manually manage the chat history.

Assistants API: For more sophisticated virtual assistants with integrated tools, context persistence, and advanced processing capabilities.

Feedback and questions

Presentation’s time is Friday at 15:00 (CET)

Do you have any idea for a project?