



# INFO-H512 : Current Trends in AI

Guillaume Wafflard

ChatACE : answering questions on the rules of a student association using Large Language Models

# 1. Introduction

# Main idea

- ChatACE : an application to ask and receive answer on the ruling of the Association des Cercles Etudiants (ACE)

- A student can ask any question about those rules

- A language model will generate an answer based on the provided documents

Link to the app web page : https://chat-ace.streamlit.app/

Link to Github repository : https://github.com/gwafflar/chatACE

# The context

- The Association des Cercles Etudiants (ACE) is a collective of “Cercles Etudiants” at ULB.
- Composed of 31 Cercles-membres
- Legal ASBL
- 37 pages of Statutes
- 27 pages of Réglement d’Ordre Intérieur (ROI)
- 6 Charters (ecology, alcool, etc)

Total : 37 pages of Statutes, 50 pages of internal rules + Annexes

⇨ issue : very few students actually know all these important rules

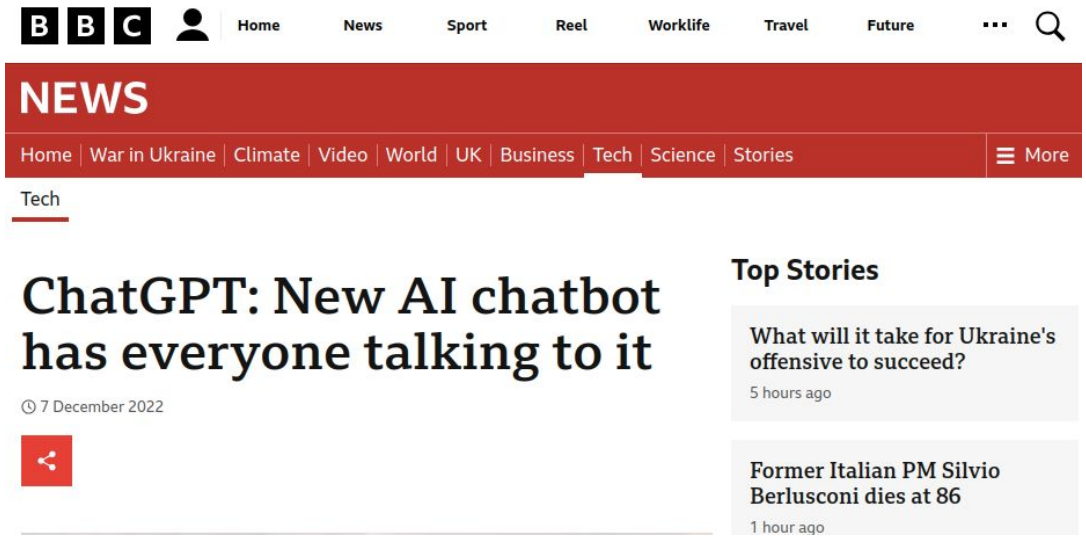
# The inspiration

Seminar : “*How can AI help law ? How can law help AI ? “* Prof. Gregory Lewkowicz

- Laws are currently written in legal language
- Difficult to understand for most people, even language natives.
- AI could help citizens to better understand legal texts

# Large Language Models (LLM)

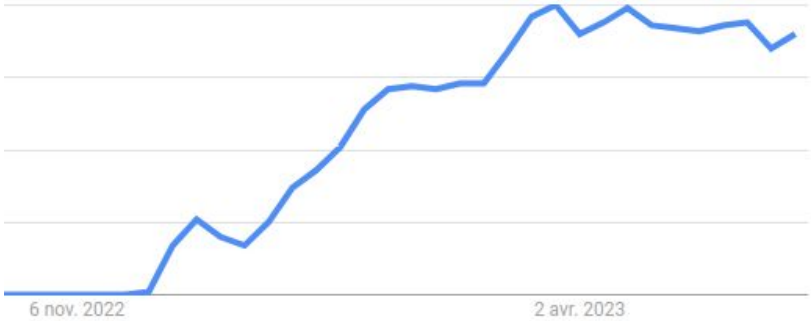
- Since the end of 2022, huge hype on chatGPT



# A current trend in AI…

- Since the end of 2022, huge hype on chatGPT
- Rise of model languages
- Revolutionary chatbot
- Used by the general public
- 100 million monthly users in less than 6 months
- March 2023 : GPT4
- State of the art in  Large Language Model

Google trends : “ChatGPT”



# The idea

Statutes and internal rules of ACE → legal documents

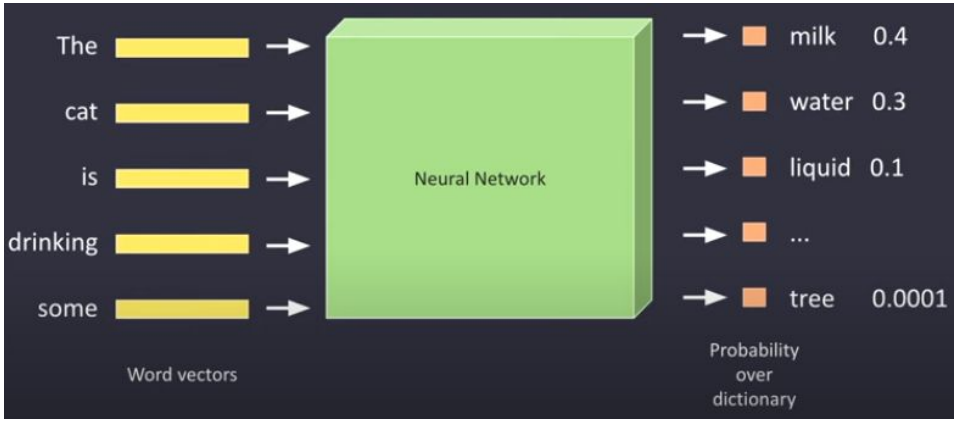
Difficult to know entirely → AI can help

Use of a current trend in AI to resolve a *real* problem

# 2. Large Language Models

# 2. Large Language Model : Definition

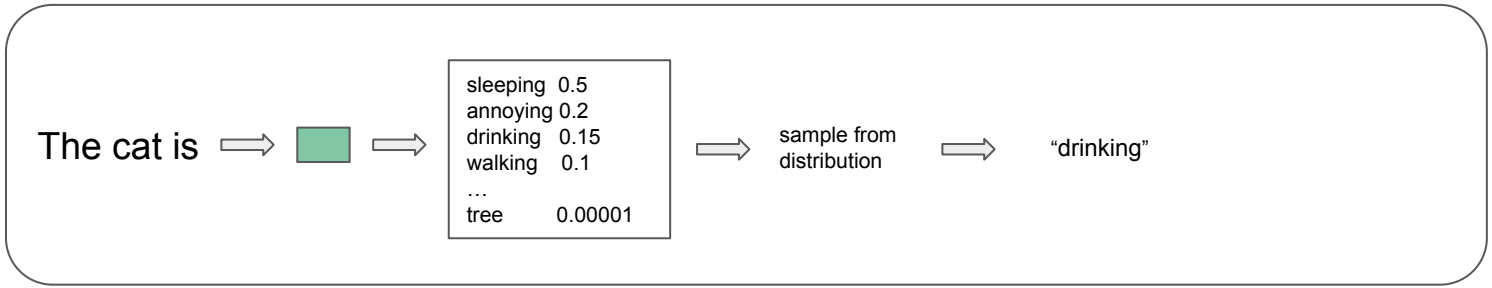
- Generative AI (also used in art)
- Deep learning : Billions of parameters
- Neural Network : transformers
- From a sequence in input, output the probability of the next token
- Generate a sequence of tokens, one after the other




From https://youtu.be/3Fp_fMtk79U
Democratizing large language models
Armand Joulin, Research Director at META

# 2. Large Language Model : Example

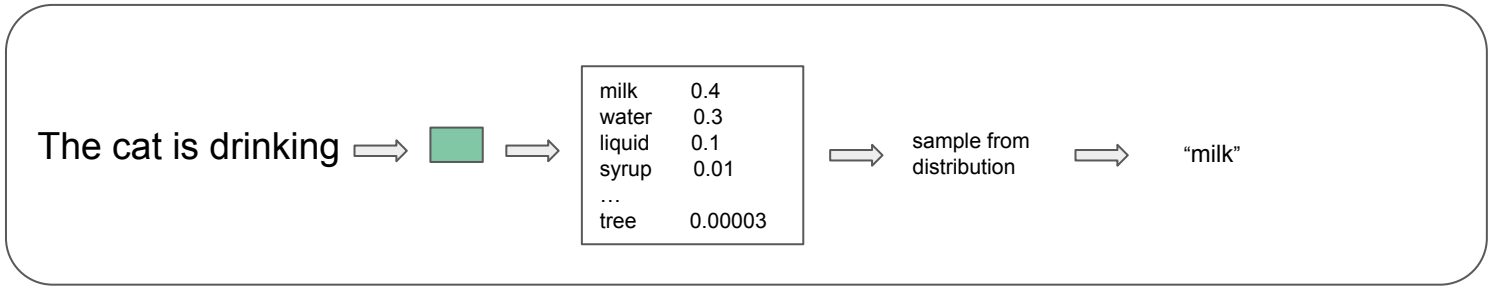
- Generative AI (also used in art)
- Deep learning : Billions of parameters
- Neural Network : transformers
- From a sequence in input, output the probability of the next token
- Generate a sequence of tokens, one after the other




Example from Armand Joulin, Research Director at META

# 2. Large Language Model : Example

- Generative AI (also used in art)
- Deep learning : Billions of parameters
- Neural Network : transformers
- From a sequence in input, output the probability of the next token
- Generate a sequence of tokens, one after the other




Example from Armand Joulin, Research Director at META

# 2. Large Language Model : Training

- Larges ML → from **billions of parameters** to **hundreds** of billions (and more…)
- Require long training (trillions of token), take months → expensive
- Requires high quality data : Wikipedia, book, scientific articles, …
- Fine-tuning : specialize a model on a specific range field

# Selected models

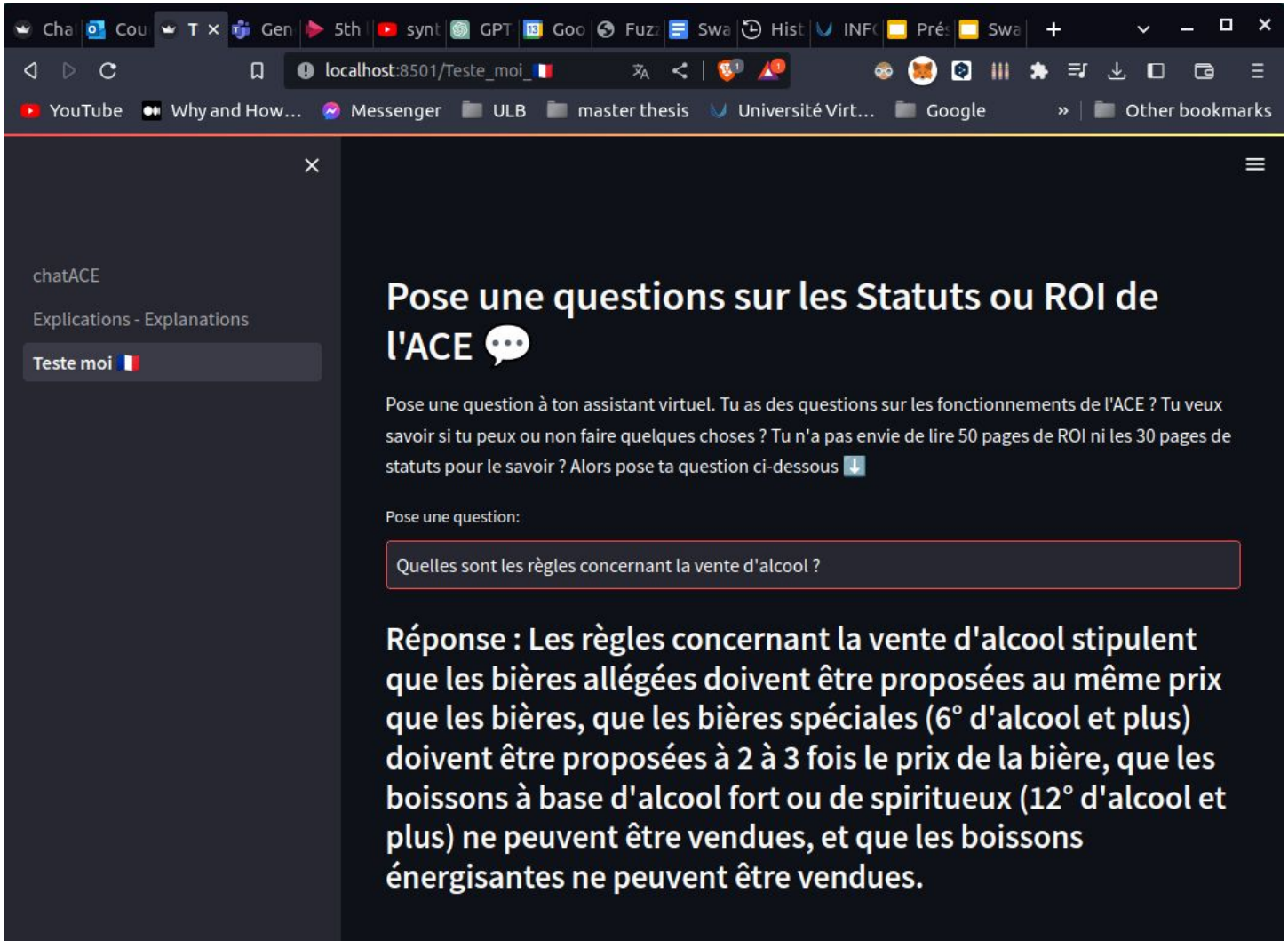
1
GPT4 (OpenAI)
Proprietary, multi-purpose

2
BIOOM (BigScience)
Multilingual, Open Source

3
Vicuna
Fine-tuned Meta's LLaMa into chatBot

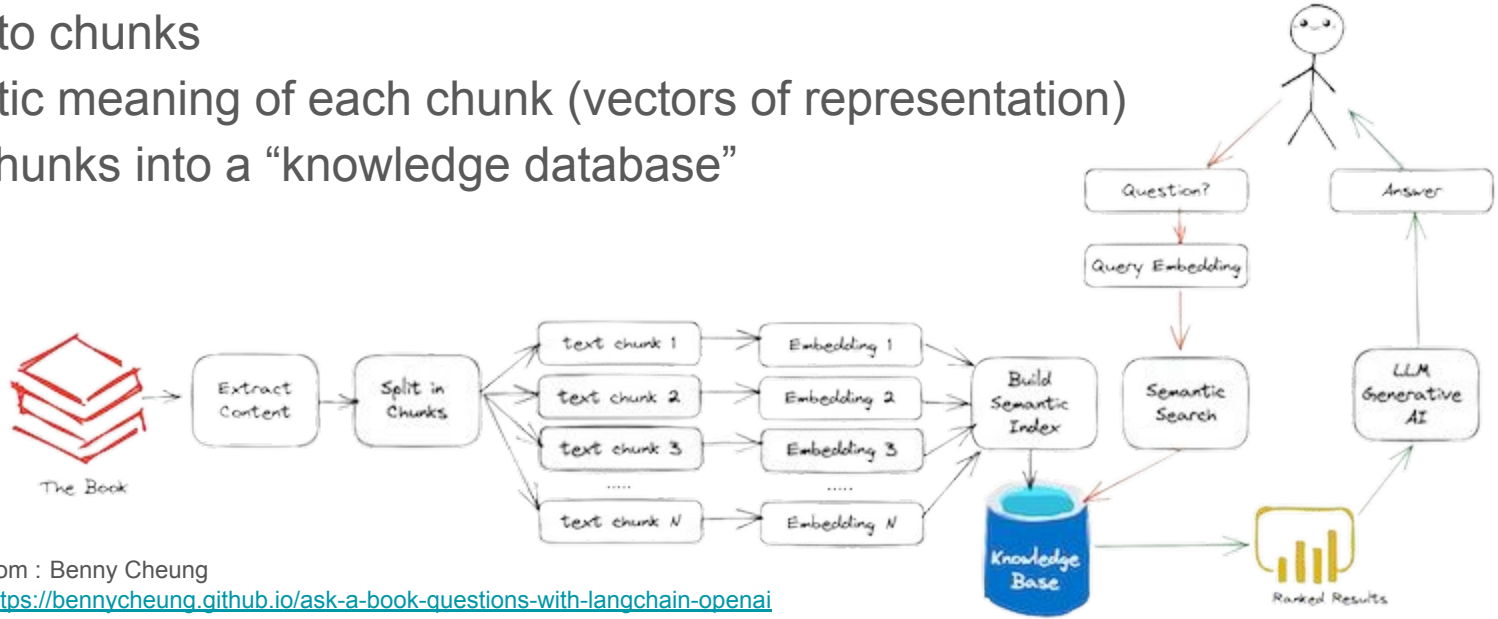
Accessed via APIs

# 3. The application


localhost:8501/Teste_moi_🇫🇷
YouTube
Why and How...
Messenger
ULB
master thesis
Université Virt...
Google
Other bookmarks
chatACE
Explications - Explanations
Teste moi 🇫🇷
Pose une questions sur les Statuts ou ROI de l'ACE 💬
Pose une question à ton assistant virtuel. Tu as des questions sur les fonctionnements de l'ACE ? Tu veux savoir si tu peux ou non faire quelques choses ? Tu n'a pas envie de lire 50 pages de ROI ni les 30 pages de statuts pour le savoir ? Alors pose ta question ci-dessous ⬇️
Pose une question:
Quelles sont les règles concernant la vente d'alcool ?
Réponse : Les règles concernant la vente d'alcool stipulent que les bières allégées doivent être proposées au même prix que les bières, que les bières spéciales (6° d'alcool et plus) doivent être proposées à 2 à 3 fois le prix de la bière, que les boissons à base d'alcool fort ou de spiritueux (12° d'alcool et plus) ne peuvent être vendues, et que les boissons énergisantes ne peuvent être vendues.

# Pre-process

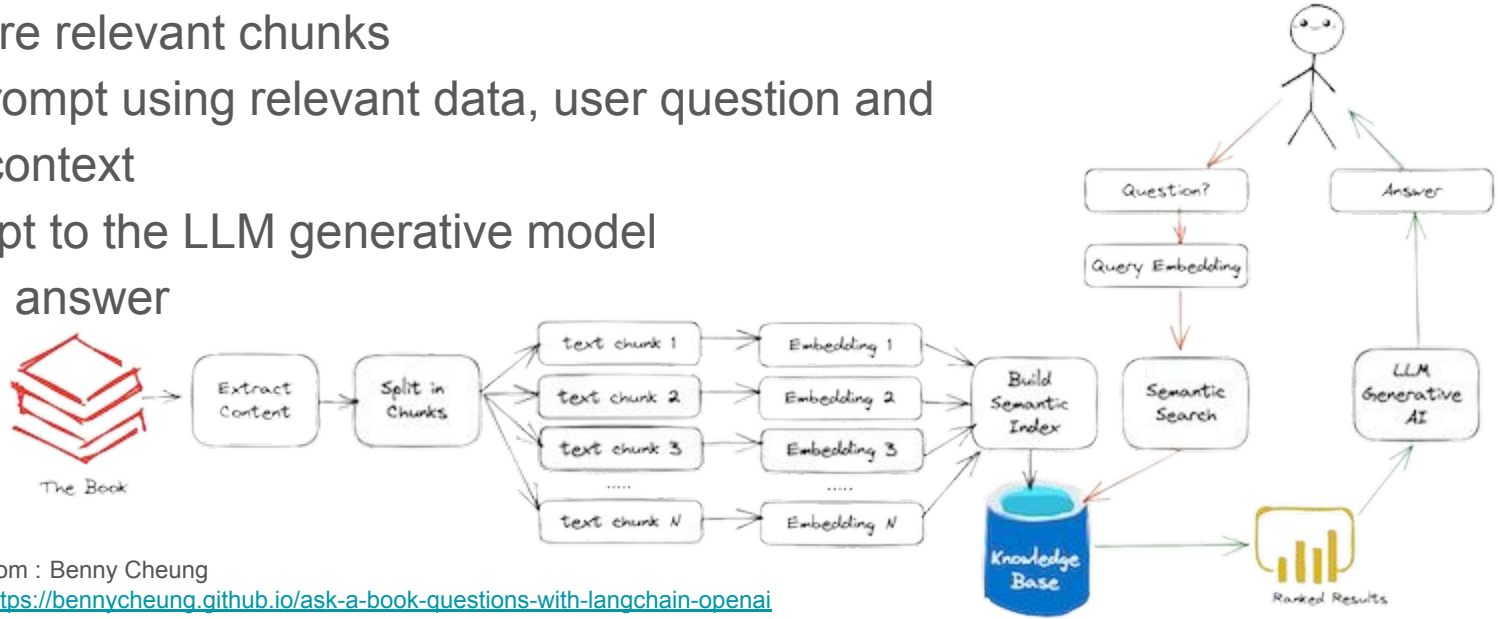
1. Get the data files (PDF)
2. Extract text
3. Clean the text
4. Split text into chunks
5. Get semantic meaning of each chunk (vectors of representation)
6. Store the chunks into a “knowledge database”




from : Benny Cheung
https://bennycheung.github.io/ask-a-book-questions-with-langchain-openai

# Process questions

1. User asks a question
2. Get the semantic meaning of the question
3. Semantic search in the “knowledge database”
4. Get the more relevant chunks
5. Create a prompt using relevant data, user question and additional context
6. Send prompt to the LLM generative model
7. Receive an answer




from : Benny Cheung
https://bennycheung.github.io/ask-a-book-questions-with-langchain-openai

# The prompt

LLM necessitate well formatted input prompt

1. **GPT4** (easy)
   - Well fine-tuned chatBot, made to manipulate text and answer questions
   - "*Using these information, answer this question*"
   - Langchain uses chain to generate prompt internally

# The prompt

LLM necessitate well formatted input prompt

2. **Vicuna**

- LLaMA fine-tuned as a chatBot (less than GPT4)
- Need context and clear instructions
- *“Tu es mon avocat. Je te fournis un texte de règlement. Sur base de ce texte, réponds à ma question que je te poserai après. Voilà mon texte : ”* + **chunks** + *”Sur base de ces informations, réponds à cette question, en citant les articles dont il relève : ”* + **question**

# The prompt

LLM necessitate well formatted input prompt

3. **Bloom**

- Not developed as a chatBot
- Inference : generate the most probable words. Do **not** answer questions or follow instructions
- Must guide it so that the generate text resemble to a continuous text
- "*Dialogue entre Alan et Marie. Marie a accès aux informations suivantes dans le règlement : Début des informations : "* + **chunks** + *"- fin des informations à disposition de Marie. Alan pose la question : "* + **question** + *" Marie lui répond naturellement : "*
- The answer looks like the continuation of the dialogue

# Generalization

The same process can be applied to *any** PDF

1. Extract text
2. Basic (not personalized) cleaning
3. Extract meaning and store as chunks
4. User asks a question
5. Retrieve chunks with similar meaning
6. Use these data to generate an answer with a language model

It is a more general extension of our specific use case.

We decided to include this option in the application.

(*just has to contain coherent text we can ask questions about)

# Live demo