LLM-ize your apps 101

AI heroes

Marco De Nittis

marco.denittis [a] gmail.com

mdn

whoami

- Independent Cloud Architect
- Trainer
- ❤️ cloud, serverless, devops, AI, wasm
- Curious and tinkerer

objectives

- Explore an LLM app with no framework
  - Bash scripts / curl (almost)
- Highlight fundamentals behaviours and building blocks

Old Manticore Inn

in the beginning…

Everyone uses ChatGPT!

We should do either to improve our business!

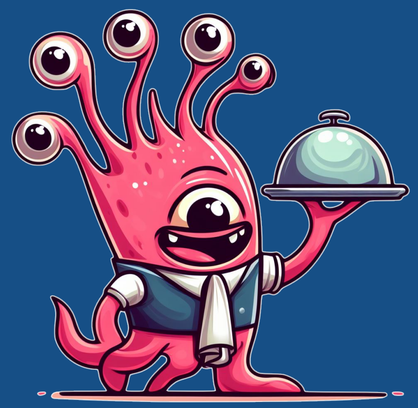
We could ask to some professionals

Too expensive!

Alex the waiter could do it!

He has a playstation and plays Fortnite every day

in the beginning…



Ok, we need Python

I hate snakes, no pythons here

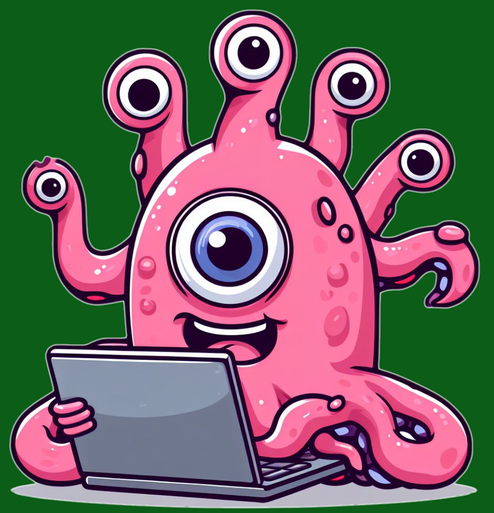
And also several cloud servers with GPUs

Ok, use this, that’s the same

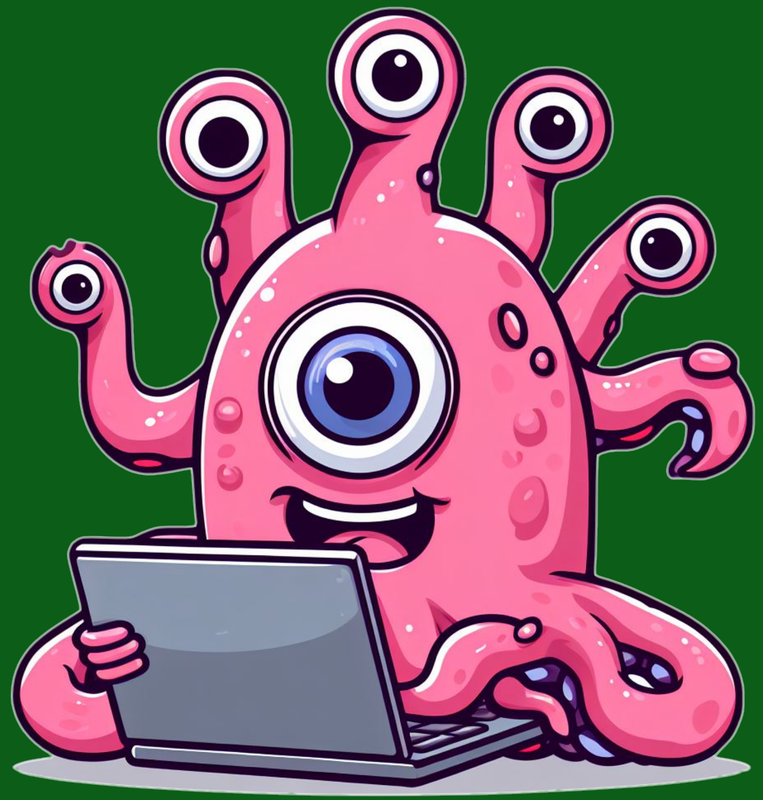
...then

first try

- Open AI API
  - Fair price
  - Rest HTTPS calls with auth token



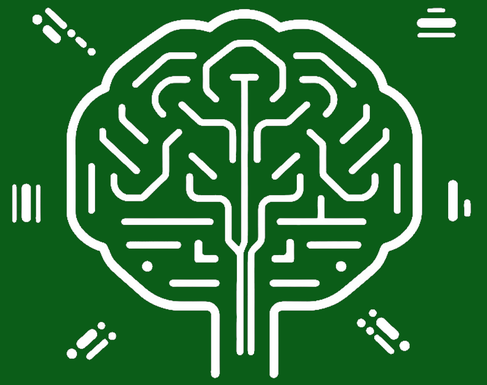
do it!



first try

- Very easy to implement

- No contextual responses



second try - context

- Prompt engineering
- Adding context with background informations
- Mixing user text with “prepared contextual text”

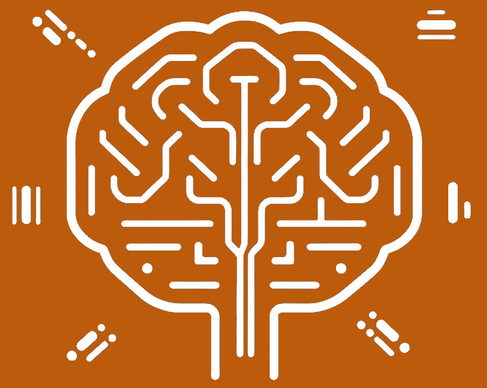
prompt engineering

- Several established techniques
  - Role: “you are an helpful restaurant assistant..
  - Shots: providing examples
  - Ask for steps

do it!

second try - context

- Tones and background are ok
- No integrations with company informations



what is a LLM

- Probabilistic engine: most probable text response from input (according to the corpus)
- Black box with input and output pipe
- No dynamic/short term memory
- Context

context

- Max input + output size
- Measured in *token*s
  - Group of letters
    - English 1.2 token/word
  - Typical size 4k - 32k, and beyond…

memory with context

- Adding all the informations in each request
- How "the chat" ChatGPT works
- Limited to context size

RAG + external memory

- Semantic memory
  - => retrieve the “most relevant” informations
- Vector database + RAG:
  - *Retrieval Augmented Generation*
- Include in context only the relevants
- Easy to edit the memory

fine tuning

- Enhancing the “*cultural baggage*” of the LLM
- Train again the model with custom specific data
- Expensive (not so much)
- Difficult to “edit” the data in memory

embeddings

- Place a text in a multidimensional “space of concepts"
- Digest the informations
  - text => vector (of floats)
- High dimensional vector (300+)
- Relevancy => mathematical distance
- A model “calculates” the embeds
- Vector database to store & search

vector database

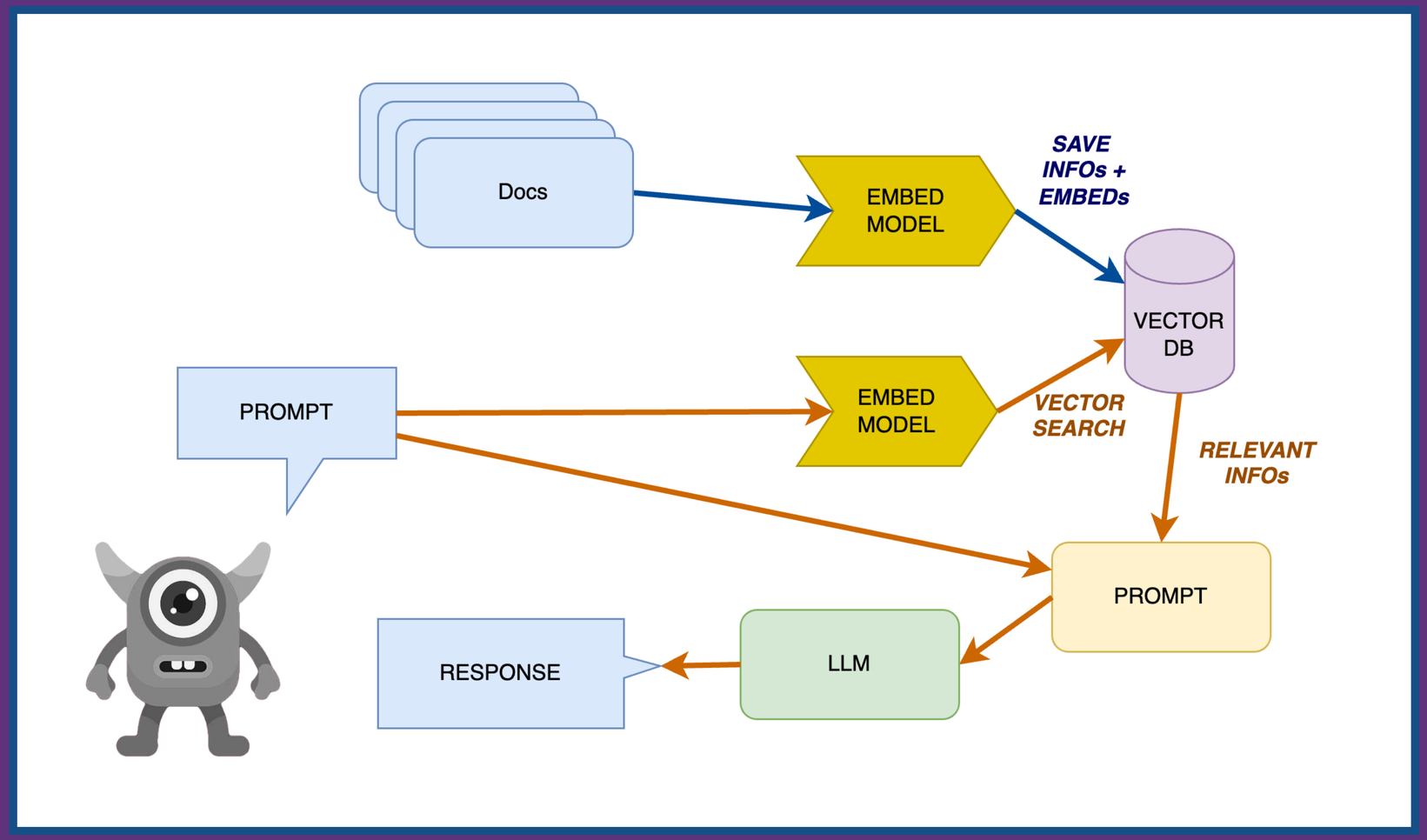
- A database capable to handle vector of floats
  - Indexing
  - Search
  - Find distances (several algorithms)
- Ad hoc db: Qdrant, Pinecone, Chroma, Milvus, …
- Almost any common db: postgres, redis, sqlite, …

phases

- Storing and indexing company info in a vector DB
  - Embeddings contents
  - Una tantum

- Embeddings of user prompt
  - Search of relevant infos
  - Embed infos in the prompt

phases


Docs
EMBED MODEL
SAVE INFOs + EMBEDs
VECTOR DB
PROMPT
EMBED MODEL
VECTOR SEARCH
RELEVANT INFOs
PROMPT
LLM
RESPONSE

do it!

improvements

- Chunking input for large document
- Check for prompt injection
- feedback (or self feedback) for the response

improvements

- Integrate LLM framework
- Assistant base approach
  - An LLM coordinating the work of other LLMs
  - Needs a full-fledged framework

point of attentions

- Control costs
- Input control
- Predictability
- Sustainability

AI heroes

thank you!

https://github.com/mdnmdn/aiheroes-2023

*https://speakerscore.it/**AIHEROES-101***

Marco De Nittis

marco.denittis [a] gmail.com

mdn