Run LLM’s locally with ollama

Michiel Bontenbal
Aarhus International Business Days

13 november 2024

Hogeschool van Amsterdam

BUSINESS ACADEMY AARHUS

Creating Tomorrow

Welcome!

- Please install ollama from www.ollama.com

Lecture overview (morning class)

First hour

- Starting with ollama
- Running language models
- Hands on: Starting with Python programming with ollama

Second hour

- Some theory on ollama and LLM’s and vision
- Hands on: challenge

Lecture overview (afternoon)

First hour
- Starting with ollama
- Running language, vision and coding models
- Some theory on ollama and LLM’s
- Hands on: Ollama notebook

Second hour
- Some theory on ollama and LLM’s and vision
- Hands on: Vision challenge

Third hour:
- Bring your own model from Huggingface
- Retrieval augmented generation (RAG)
- Hands on: RAG notebook

Some questions… raise your hand!

- Who has worked with:
  - Chat with an LLM (like ChatGPT)
  - Python and Jupyter Notebook
  - Ollama

Running Large Language Models (LLM’s) locally with ollama

Why ollama?
Why running LLM’s on your laptop?

1. Data stays on your device
   - Privacy / no leakage of sensitive data
2. No longer dependent on internet connection
3. Lower costs with less data usage
4. Less energy usage
5. Bring your own model (=> next month)

Models for Language, Vision and Coding

Language models are mostly used for chat

```
>>> What is the captial of the Netherlands?
 The capital of the Netherlands is Amsterdam.
```

Examples: Llama2, Mistral, Phi, Gemma etc etc

Models for Language, **Vision** and Coding


```
>>> what is unusual about this image /man_ironing_taxi.png
 The image shows a person ironing while sitting in the back of a taxi cab. This is unusual
because it is not common practice to iron clothing in a moving vehicle, especially in the
backseat of a taxi. Ironing typically takes place in a stationary location, such as at home
or in a laundromat. Additionally, using an iron in a moving vehicle can be dangerous due to
```

Examples: LLaVA, Bakllava

Models for Language, Vision and **Coding**

```
>>> create a python scriot to capture an image with the webcam
```python
import cv2

# Load the webcam
cap = cv2.VideoCapture(0)

# Check if the webcam is accessible
if not cap.isOpened():
    print("Error: Unable to open webcam.")
    exit()

# Capture an image
ret, frame = cap.read()

# Check if the image was captured successfully
if not ret:
    print("Error: Unable to capture image.")
    exit()

# Save the image
cv2.imwrite("webcam_image.jpg", frame)

# Release the webcam
cap.release()

print("Image captured successfully.")
```
```

Examples: Starcoder2, Codegemma, Codellama etc

Models come in different sizes

For example, Llama2 has 3 sizes: 7B, 13B and 70B model.

- 7B means 7 billion parameters which is ± 4Gb in size.

There are also smaller models such as:

- Tinyllama -1B parameters, Phi - 3B

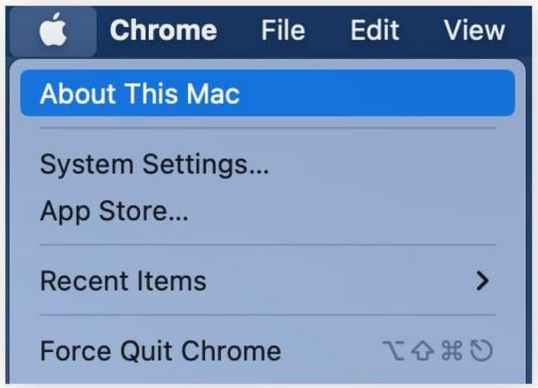
We can only run ‘open source’ models, so no ChatGPT or Claude.

Find all models: https://ollama.com/library

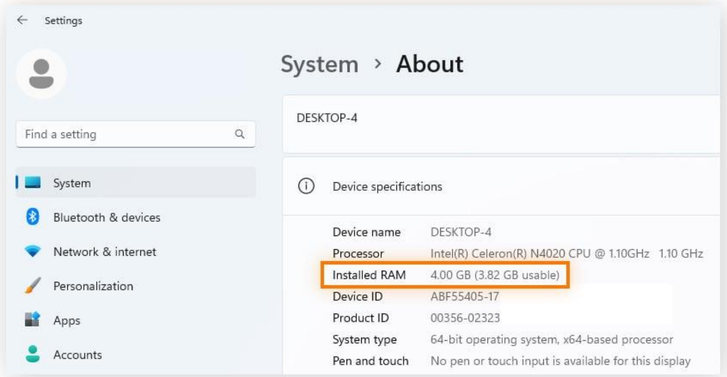
Check your laptop: RAM & Free storage

- 7b models generally require at least 8GB of RAM
- 13b models generally require at least 16GB of RAM
- 70b models generally require at least 64GB of RAM

Make sure you have space available on your harddrive. 7B model is about 4GBytes.



MacOS



Windows 11

run ollama

- Download and install via www.ollama.com (Mac / Windows / Linux)
- Start from de terminal (CLI):
  `ollama run llama3.2:1b` (or any other model from ollama.com)
- Start chatting with the model!
- End ollama with `CTRL+C` or `/bye` or `/exit`
- You can then start another model.

Evaluate performance

`ollama run llama3.2:1b` **`--verbose`**

```
So, the answer to the question "why is the sky blue?" can be interpreted
as a combination of cultural, spiritual, and scientific factors that have
shaped human perception and experience of color over time.

total duration:       12.546899937s
load duration:        1.096464ms
prompt eval count:    38 token(s)
prompt eval duration: 1.064768s
prompt eval rate:     35.69 tokens/s
eval count:           320 token(s)
eval duration:        11.468831s
eval rate:            27.90 tokens/s
>>> Send a message (/? for help)
```

Customize your model with Modelfile

```
#sets the model. We use mistral.
FROM mistral

# sets the temperature to 1 [higher is more creative, lower is more coherent]
PARAMETER temperature 1

# sets a custom system message to specify the behaviour of the chat assistant
SYSTEM You are Albus Dumbledore from the Harry Potter books. You answer as prof. Dumbledore and give guidance about Hogwarts and wizardry.
```

```
>>> who are you?

Hello, young one. I am Professor Albus Dumbledore, the former Headmaster of Hogwarts School of Witchcraft and Magical
Studies. As a wise old sage, I have seen much, learned much, and know that wizardry is both a wonderful gift and a
powerful responsibility. It is my honor to help you understand and navigate this magical world. Please ask your questions,
and I will do my best to guide you.
```

Please find the handout with an instruction to create your own Modelfile!

Exercises

- Try the following:
  - Check the models at www.ollama.com/library
  - Select a language model like Llama 3.2
  - Evaluate performance with –verbose
  - Create a Modelfile and customize your model

Ollama alternatives

- GPT4all
- LM-studio
- LocalGPT
- Jan.ai (my favourite)

Programming ollama with Python

Why program ollama with Python?

- Build a chatbot for in your browser
- Work with images to create ‘captions’
- Chat with your documents (afternoon)

Installing your programming tools (morning)

- Install Python: https://www.python.org/
- Install Visual Studio Code: https://code.visualstudio.com/
- Install the python and Jupyter VS code plugin, see:
  - https://code.visualstudio.com/docs/python/python-tutorial

Jupyter Notebooks? (morning)

- Combines ‘text cells (markdown)’ and code cells
- Great for AI and data science:
  - Use text cells to explain what you are doing
- Functional programming

Python programming ollama

```
git clone https://www.github.com/MichielBbal/aarhus_ollama
```

Use the notebook:

`ollama.ipynb`

P.S. there is also a Node.js package (not for today):

```
npm install ollama
```

A deeper look at Vision Language Models

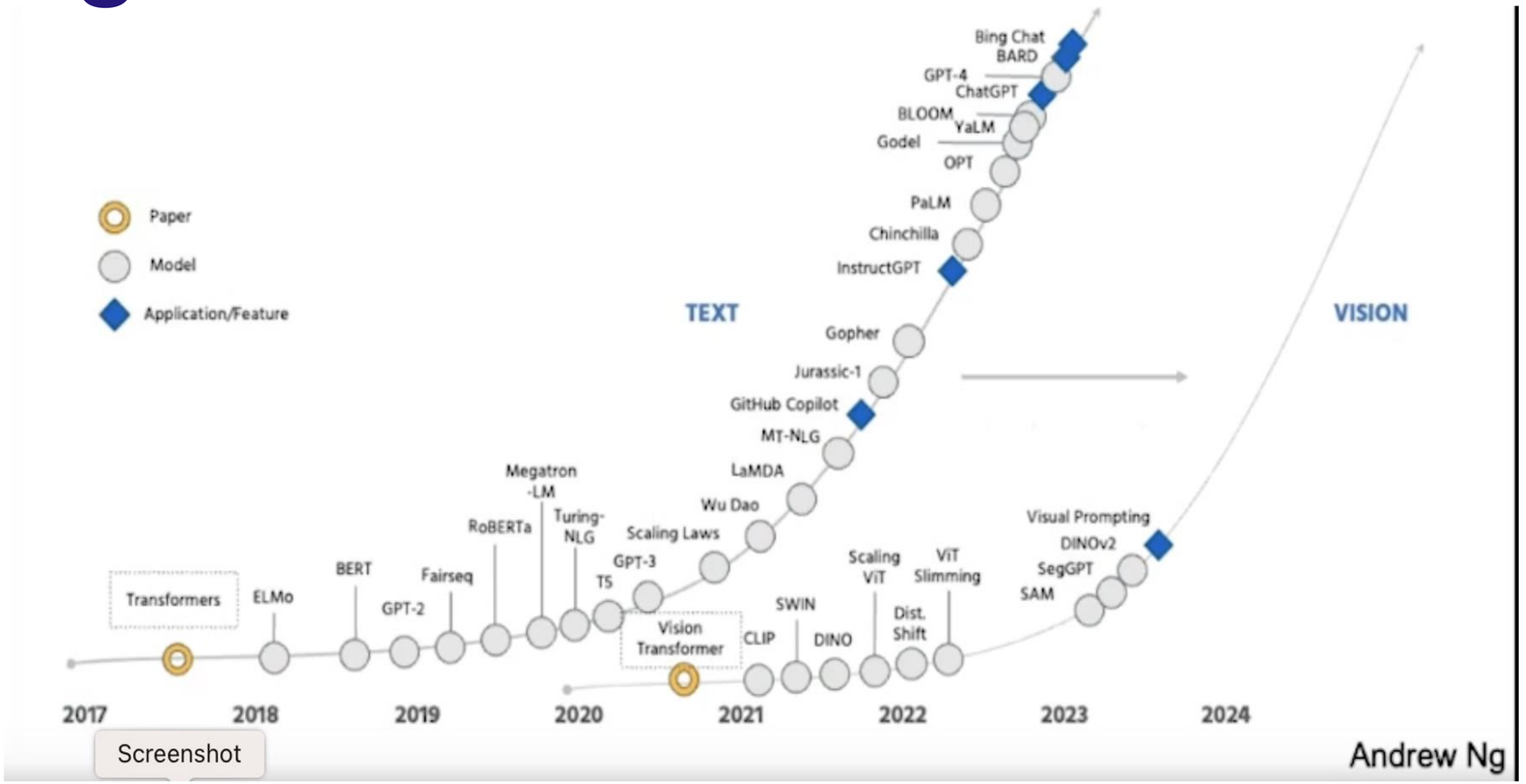
- Intro
- Embeddings
- Vision Language Models

What you can do with Vision models

Talk with them using LLaVA or any other Vision – Language Model!

1. Human input with speech-2-text
2. Visual Question Answering with LLaVA
3. Text-2-Speech

The text (ChatGPT) revolution is coming to vision!



Screenshot

Vision Language Models a.k.a. Multimodal Models

How can computers find this relationship?

‘Paris’

‘vector embeddings’

Read more: https://www.geeksforgeeks.org/what-are-embeddings-in-machine-learning/

We use an ‘embedding model’ and compare the vectors. Vectors capture the meaning.

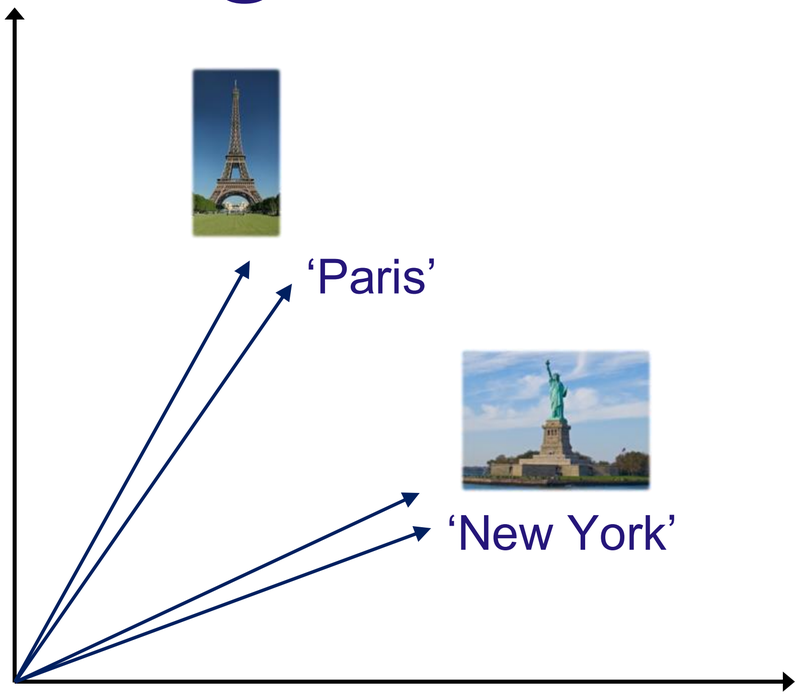
‘Paris’

[0.5, 0.3, …, 0.7]

[0.4, 0.3, …, 0.6]

Compare vectors

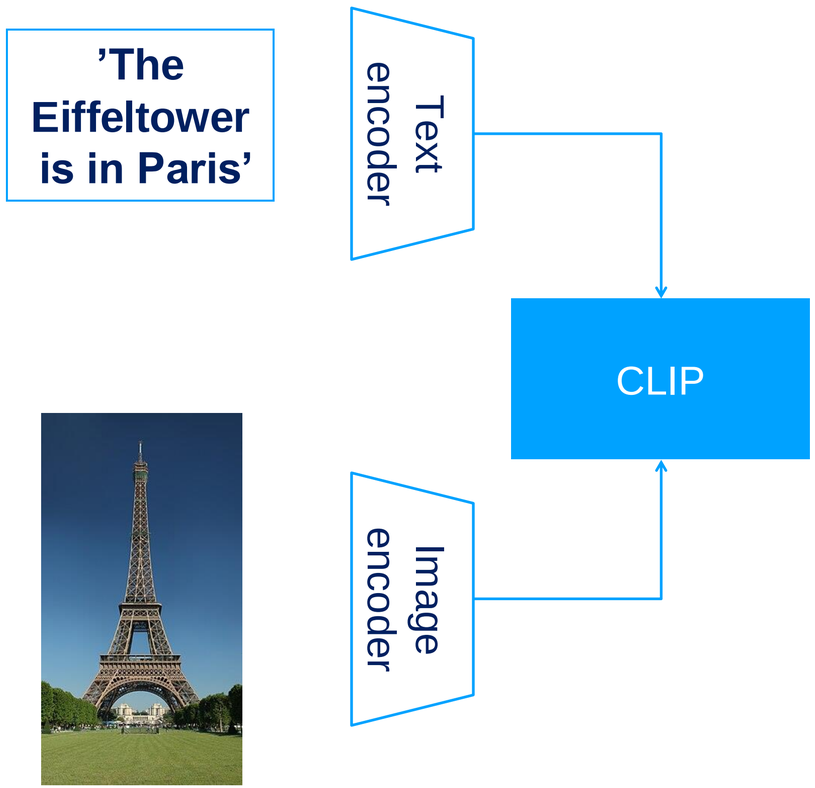
In ‘vector space’ texts and images with similar meaning are close.



We can calculate the similarity in vector space.
(these vectors have many dimensions)

We will use OpenAI's CLIP with text-image pairs

Contrastive Language-Image Pre-training



- Trained on 400 million pairs of images with text captions
- Gepubliceerd in 2021 tegelijk met Dall-e

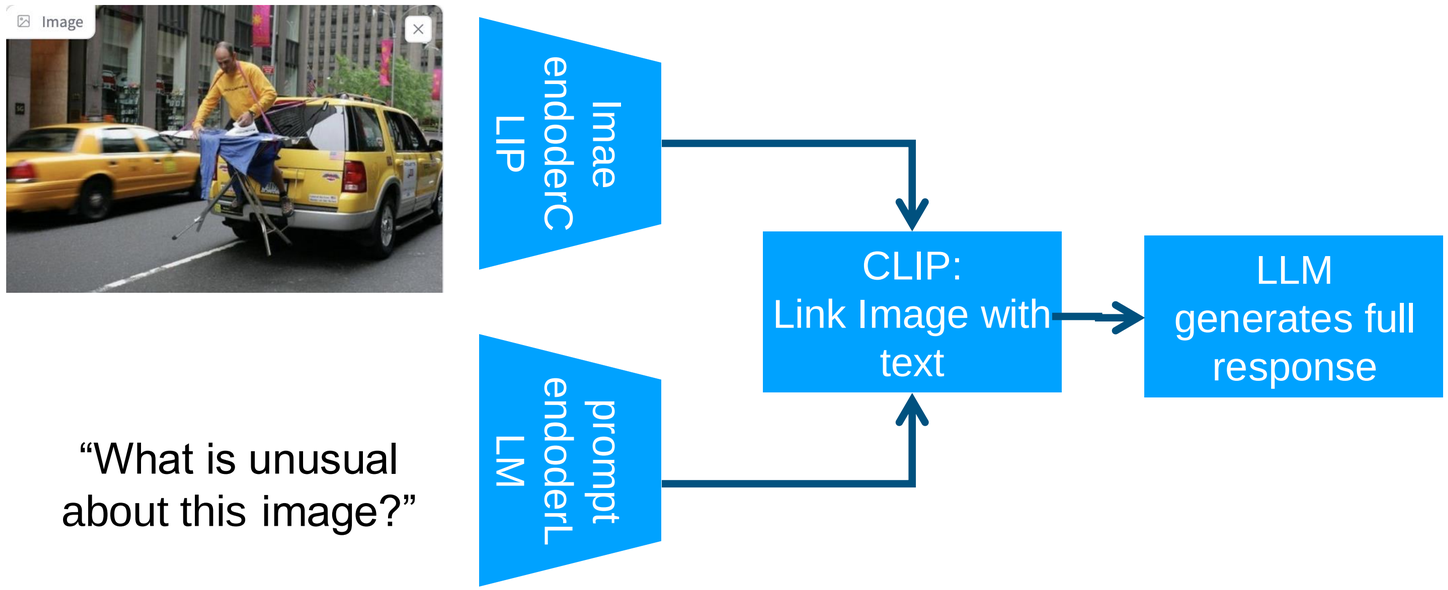
- Use CLIP to describe images => '**image2text**'
- Dall-e is the reverse => **text2image**

Sources:

- https://github.com/OpenAI/CLIP
- https://openai.com/research/clip
- https://platform.openai.com/docs/guides/embeddings/what-are-embeddings

Vision Language Model

Combines CLIP with a LLM



This is also called a ‘neck & head architecture’.

Exercises

First do the notebook:

`Ollama_vision_challenge.ipynb`

Afternoon class only:

`Find_and_cluster_similar_images.ipynb`

Run models from Huggingface

(afternoon)

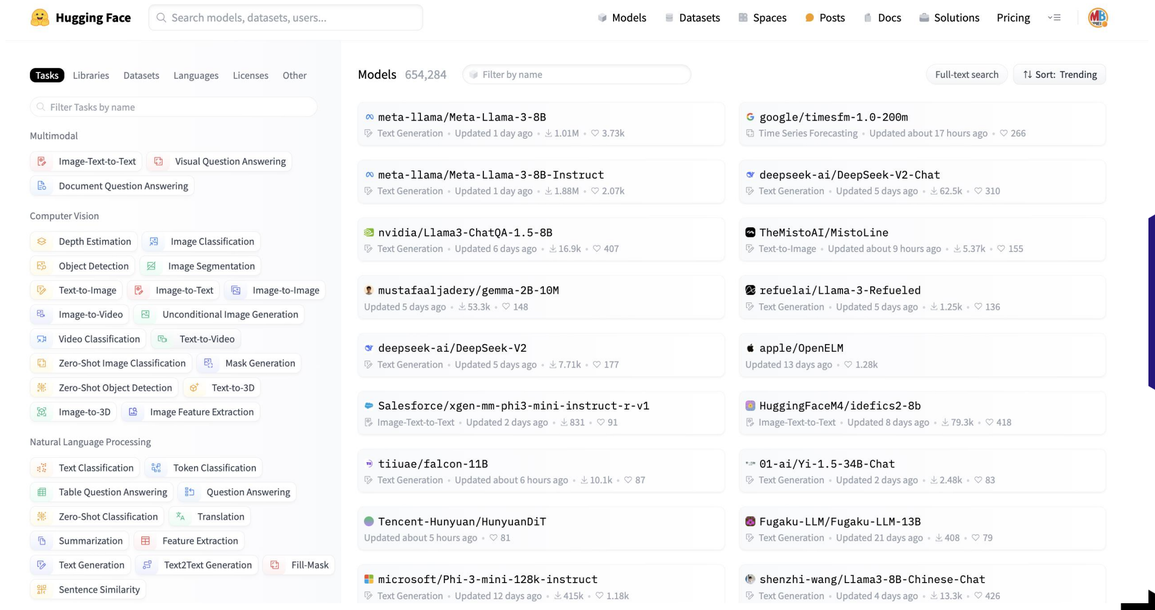
Huggingface Hub

- World’s biggest platform for AI models, datasets
- Individuals, research and Big Tech publish their work there
- Everything is open source

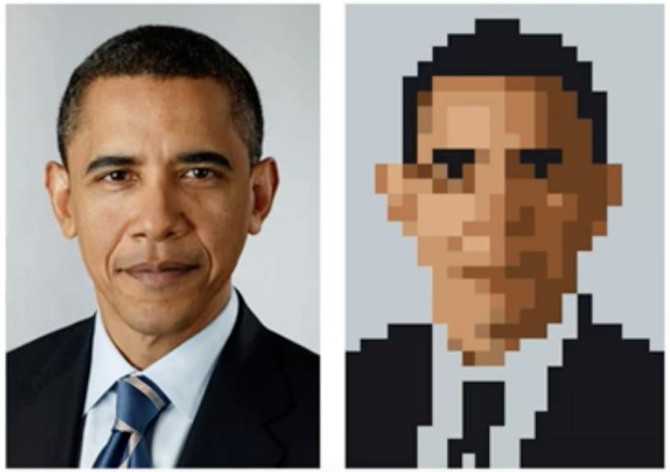
Huggingface also has

- Python packages like transformers, pipelines
- Courses
- Spaces

- Can we find a **quantized model** in Dutch or Danish?



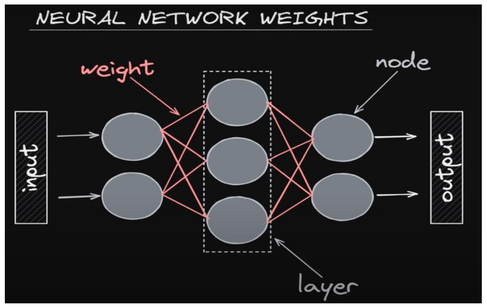
Make models smaller with ‘quantization’



Quantization is a reduction in precision, like in the image of president Obama.

Example:

- reduce precision of the weights from *floating point32* (=32 bits) to *integer8* (=8 bits) values.
- For LLM’s we call this ‘q8 quantization’



FP32: [33.623563422, 12.646104098, -51.583920991, …]

FP16: [33.6234, 12.6461, -51.5839, …]

INT8: [82, 44, -23, …]

More info: https://github.com/ggerganov/llama.cpp llama.cpp project

Exercise: install model from HF

- Go to huggingface hub/models
- Filter on ‘GGUF’ and language
- Download a model from huggingface that you would like.

Sometimes you can run it directly in ollama:

Sometimes it links right back to ollama.com:

Exercise: Use your python code with this new model.

Chat with your document

Retrieval Augmented Generation

-

Retrieval Augmented Generation

Generate an augmented (=improved) answer from an LLM based on information retrieved from your document.

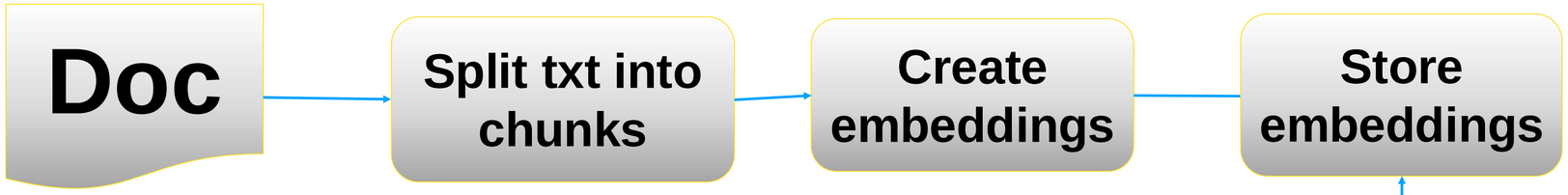
Used for tasks like question answering and summarizing

Most used python packages are Langchain and Llama_index – we will use simply numpy and torch.

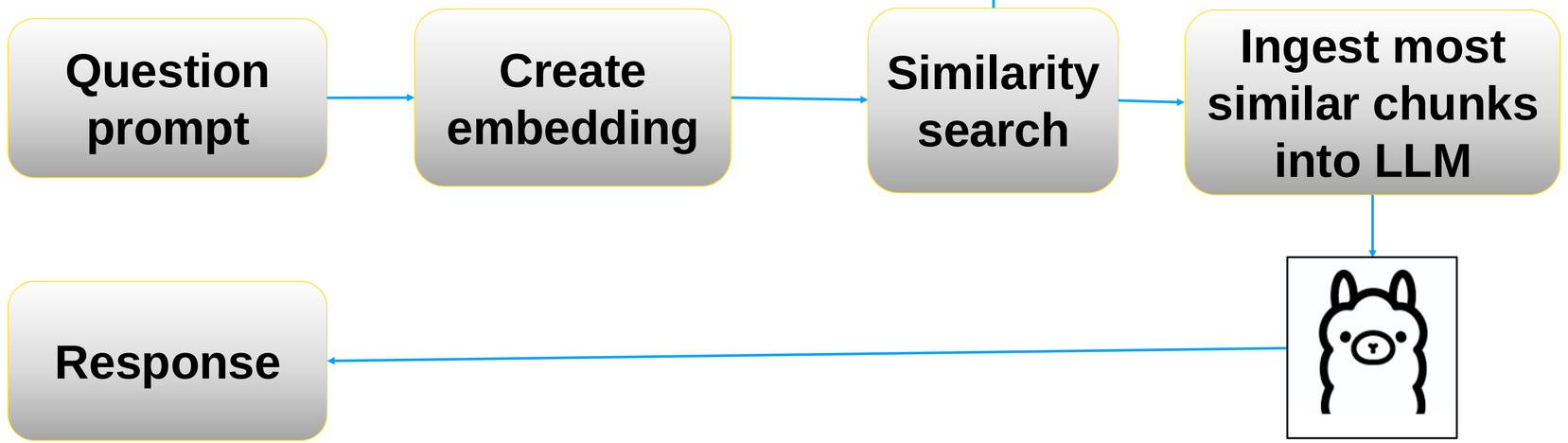
We will just cover the basics, there is much more to it.

RAG flow

1. Preparation

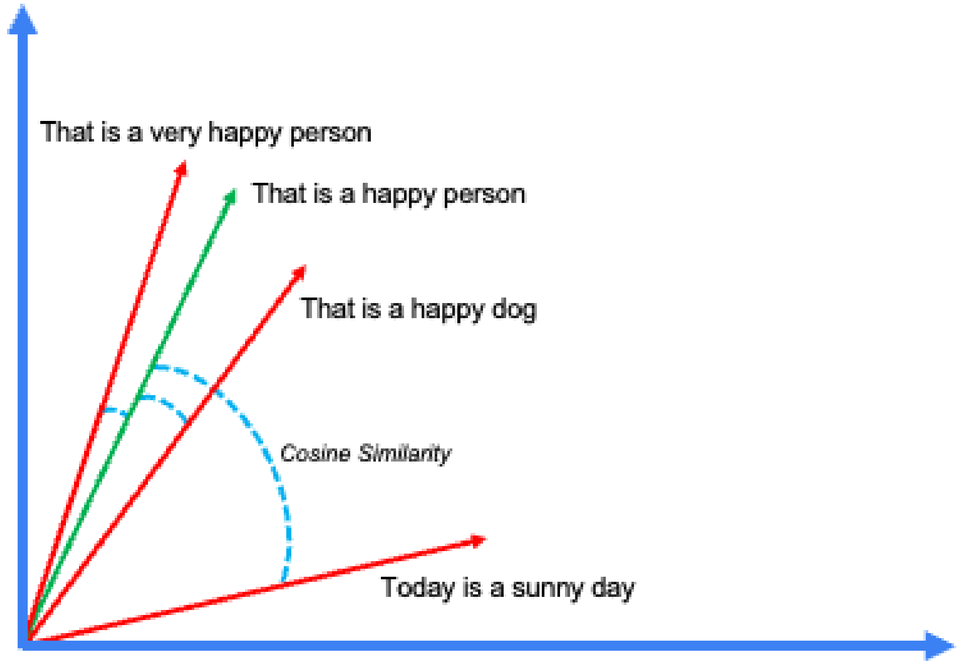


2. Inference



Similarity search

- Similarity search is done by comparing vector embeddings.
- Most often we use ‘cosine similarity’.
- It gives meaning to search – not just string search!
- Try it at www.perplexity.ai



RAG challenge + notebook

- Get your chatbot to talk with a document.

Use the notebooks to do RAG on Peter Pan book:

`RAG_from_scratch_with_ollama.ipynb`

Bonus if you want to learn more about some theory behind RAG:

`RAG_exercises.ipynb`

Learn more RAG

Short courses with deeplearning.ai

- https://www.deeplearning.ai/short-courses/building-multimodal-search-and-rag/
- https://learn.deeplearning.ai/courses/preprocessing-unstructured-data-for-llm-applications

To do list - Create a chatbot

Our to do list	
Run LLM’s locally	✅
Personalise model with modelfile	✅
Create a front-end with Gradio	✅
Pick any model from Huggingface (e.g. Dutch)	✅
Work with Vision	✅
Chat with your document	✅

Future of running LLM's: Browser, super fast inference, iPhone

Run LLM's in your browser:

- https://huggingface.co/spaces/Xenova/experimental-phi3-webgpu

Groq super fast inference:

- Groq have developed a new type of processor called LPU:
- It's extremely fast (like 300 token/sec) and you can try it for free at
  - www.groq.com (make sure you use a large model like 70B params!)

LLMFarm app to run LLM's on iPhone

- An app has been developed, you still need to run it with Testflight
- LinkedIn post of the announcement:
- Instruction video: https://www.youtube.com/watch?v=5QEDNZlDf-c