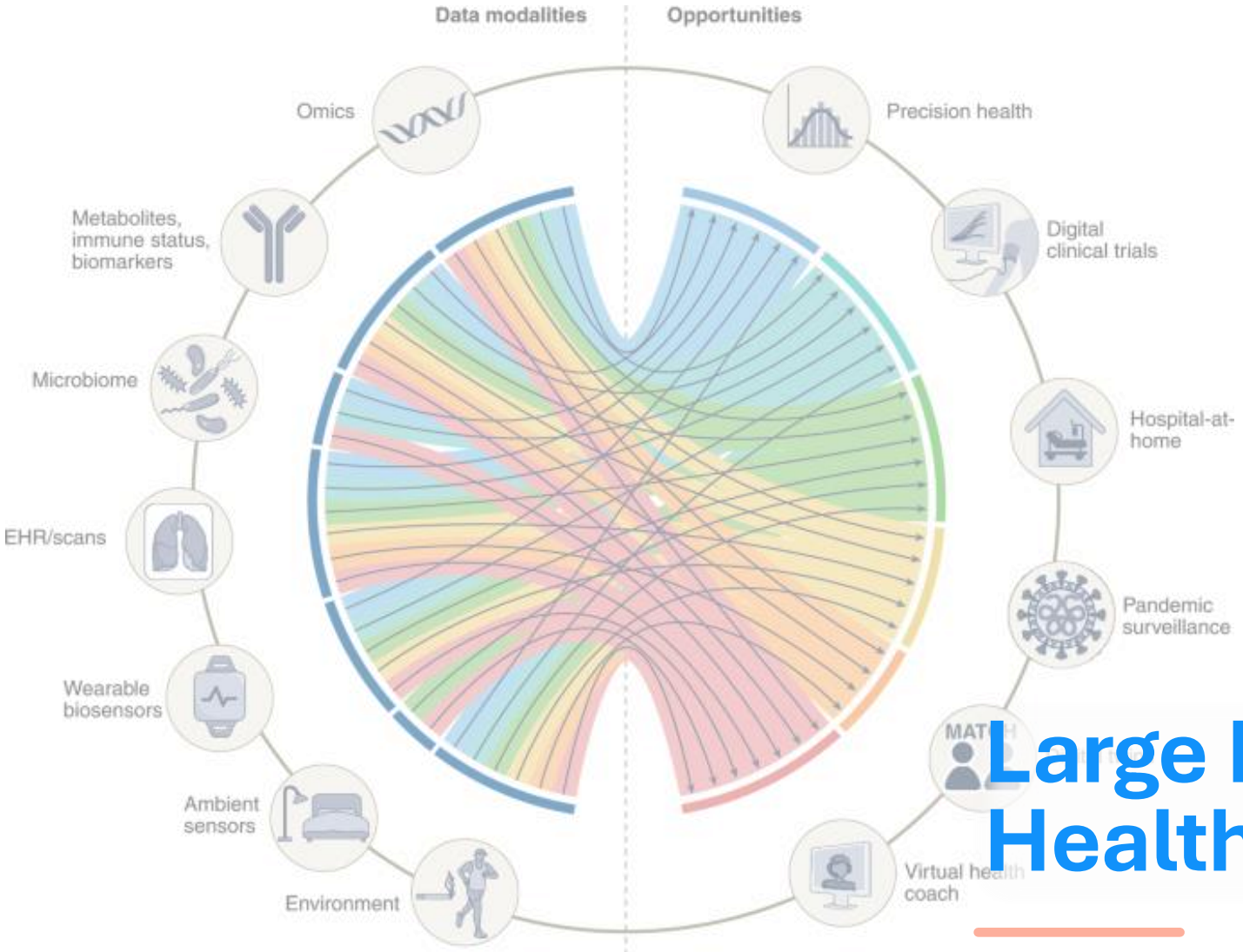


Large Language Models in Healthcare

Day 3: Multimodal healthcare models

Ruurd Kuiper – Julius Center, UMCU

UMC Utrecht

Day 3: Multimodal healthcare models

Topics covered:

1. Introduction to healthcare models
2. Validation metrics
3. Introduction to Huggingface and Kaggle
4. **Practical session:** fine-tuning a model for healthcare

Short break

4. Vision-Language Models (VLMs)
5. Exploring different healthcare specific models
6. **Practical session:** using an open-source VLM

Lunch

7. Retrieval Augmented Generation (RAG)
8. **Practical session:** Using and evaluating proprietary models
9. **Assignment:** Setting up your own validation experiment

Discuss yesterdays practical

Are there any questions?

Was it easy/do-able/hard?

Did you understand the code?

Did you understand the output?

Transformer visualization

Attention explained video

Introduction to healthcare models

What's different?

Description:
Most healthcare models are (pre-trained) models, adapted to healthcare specific tasks.

When we use or develop models for healthcare, we must make choices

Today we will be discussing various aspects, such as:

- Modalities
- Size (quantization)
- Open- vs. closed source
- General vs. finetuned models
- Validation (metrics)
- Data- and model-hubs

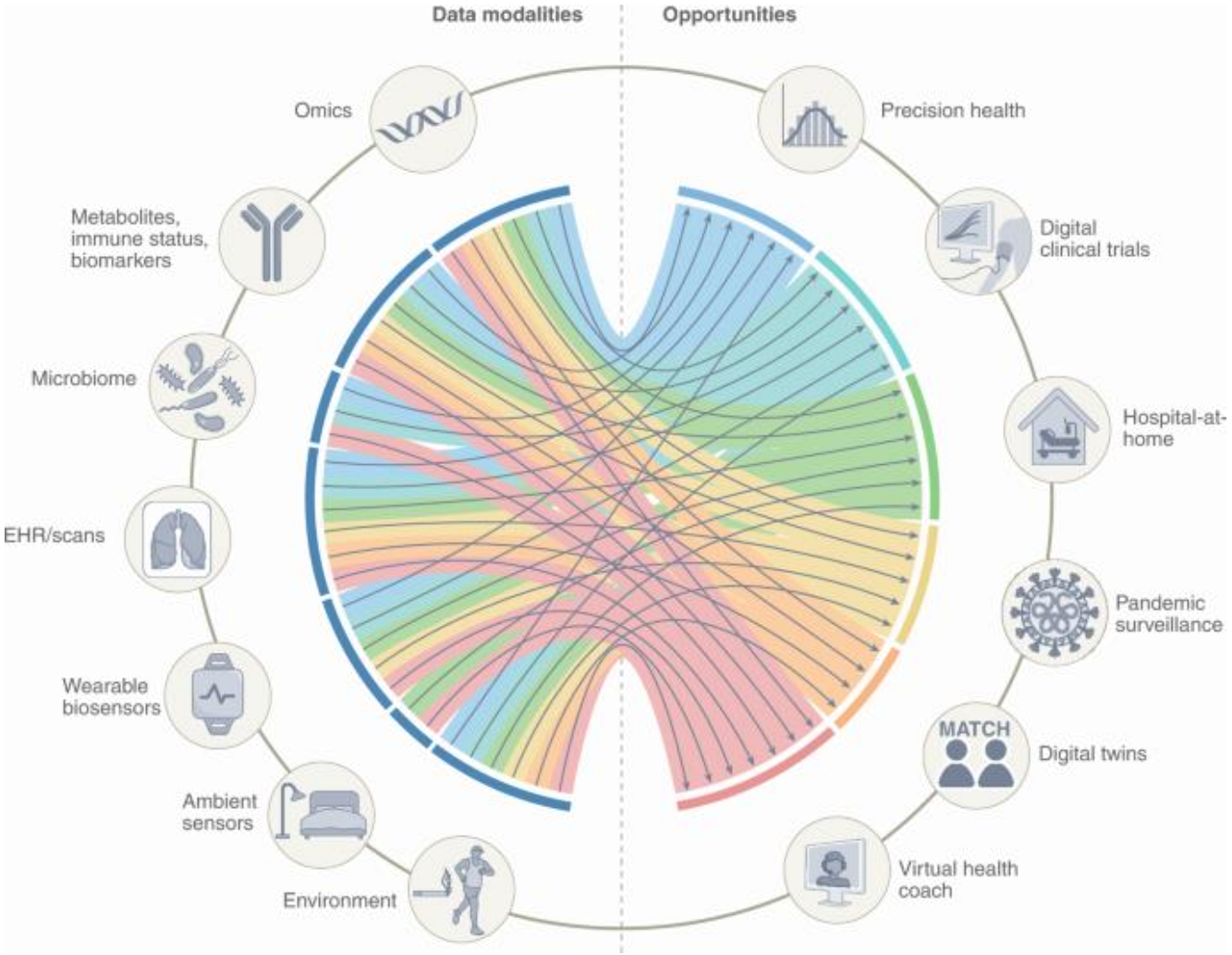
These choices will be important during the assignment!

Introduction to healthcare models

Why multimodal?

- Healthcare data is **multifaceted**
- Most healthcare data is not text!
- Structured vs. unstructured data
- **Unimodal models** miss important context
- **Multimodal models** synthesize information

Example: Chest X-ray + patient history → better diagnosis than either alone.



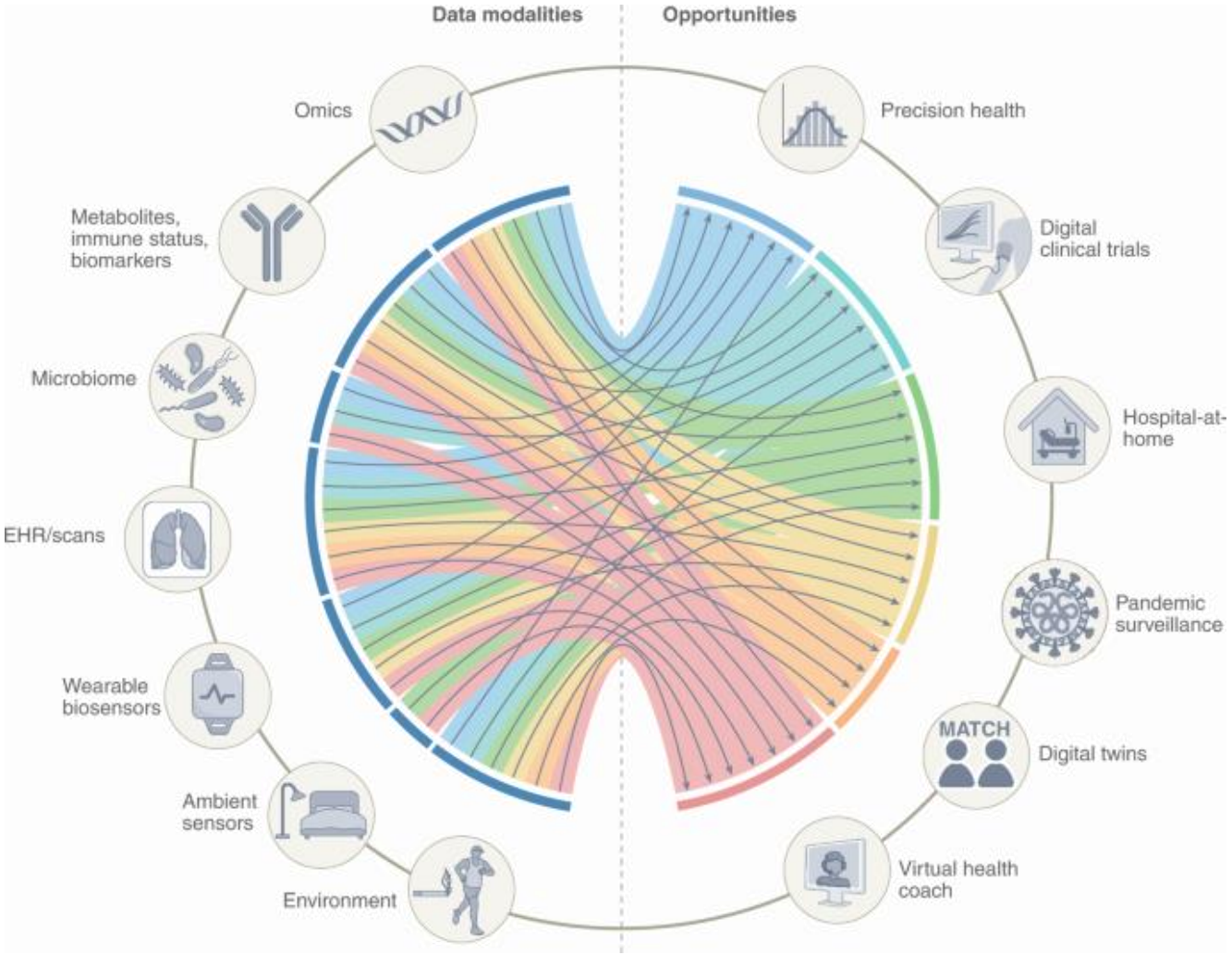
Acosta, J.N., Falcone, G.J., Rajpurkar, P. *et al.* Multimodal biomedical AI. *Nat Med* **28**, 1773–1784 (2022). https://doi.org/10.1038/s41591-022-01981-2

Introduction to healthcare models

Types of (healthcare) data

- **Text:** clinical notes, discharge summaries, radiology reports.
- **Images:** X-rays, CT scans, MRIs, ultrasound, histopathology slides.
- …

Can we think of more?



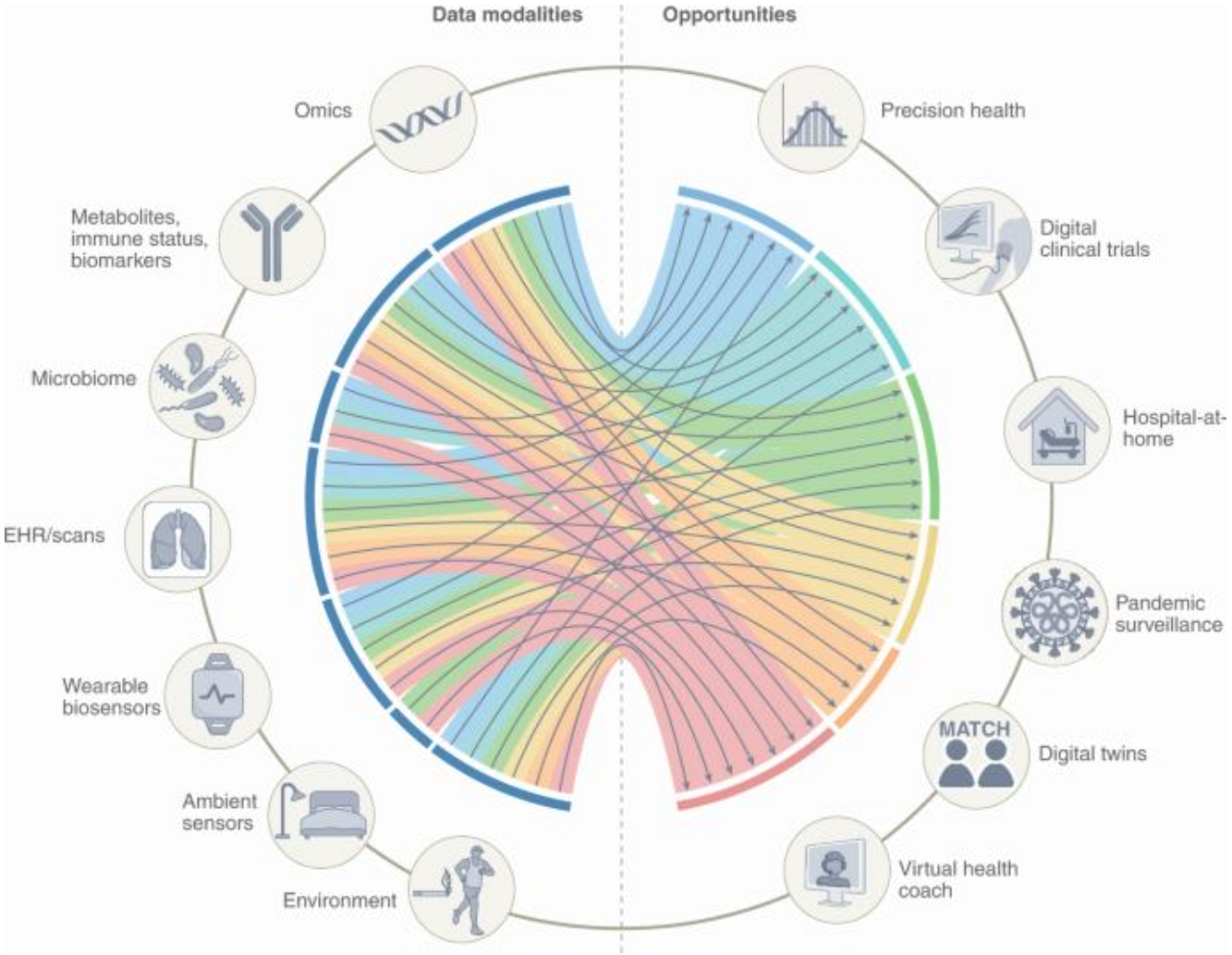
Acosta, J.N., Falcone, G.J., Rajpurkar, P. *et al.* Multimodal biomedical AI. *Nat Med* **28**, 1773–1784 (2022). https://doi.org/10.1038/s41591-022-01981-2

Introduction to healthcare models

Types of (healthcare) data

- **Text:** clinical notes, discharge summaries, radiology reports.
- **Images:** X-rays, CT scans, MRIs, ultrasound, histopathology slides.
- **Audio:** patient-doctor conversations, auscultation sounds.
- **Physiological signals:** ECG, EEG, vital signs, wearable sensors.
- **Genomics / omics data:** DNA sequences, protein expression, biomarkers.
- **Video:** endoscopy, surgery recordings, patient interaction

Can we think of more?



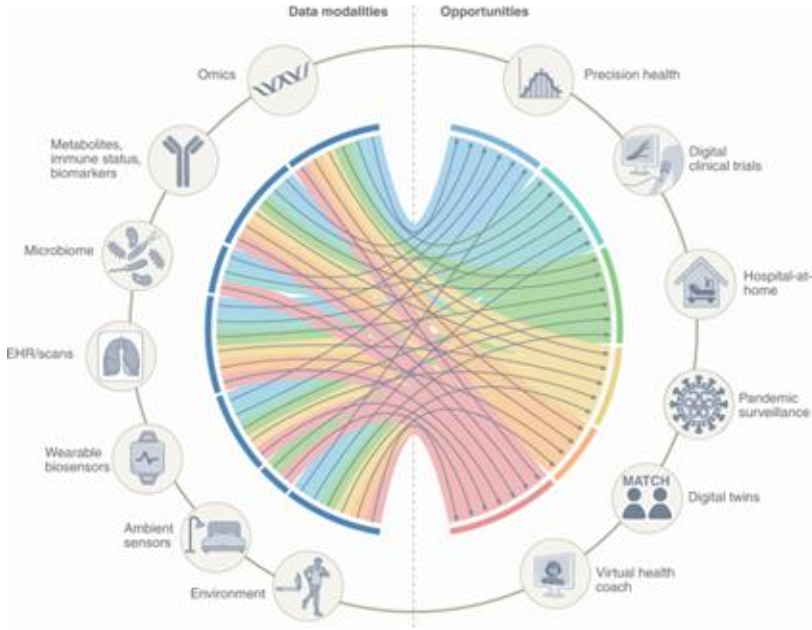
Acosta, J.N., Falcone, G.J., Rajpurkar, P. *et al.* Multimodal biomedical AI. *Nat Med* **28**, 1773–1784 (2022). https://doi.org/10.1038/s41591-022-01981-2

Introduction to healthcare models

Applications

- **Medical image interpretation**
- **Automated report generation** (radiology, pathology).
- **Clinical decision support** combining patient history + imaging.
- …

Can we think of more?

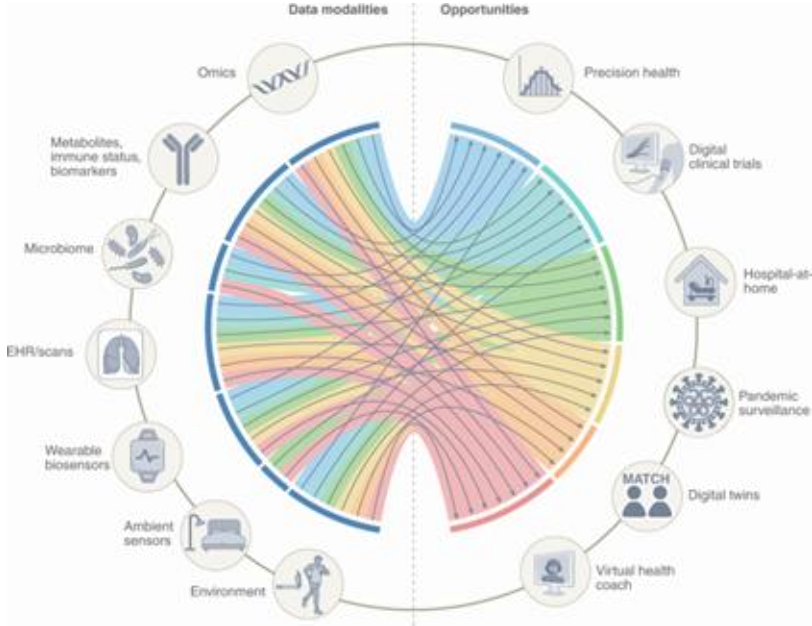


Acosta, J.N., Falcone, G.J., Rajpurkar, P. *et al.* Multimodal biomedical AI. *Nat Med* **28**, 1773–1784 (2022). https://doi.org/10.1038/s41591-022-01981-2

Introduction to healthcare models

Applications

- **Medical image interpretation**
- **Automated report generation** (radiology, pathology).
- **Clinical decision support** combining patient history + imaging.
- **Patient triage** using vitals + symptoms + text input.
- **Disease progression monitoring** from multimodal longitudinal data.
- **Transcription:** integrating audio conversations + background info.
- **Personalized medicine**: linking genomics, labs, and imaging.
- **Telemedicine**: multimodal assessment from remote inputs (video, audio, text).
- **Workflow orchestration**: Digital assistant (for PACS, EHR)
- **Research assistance:** summarizing, cleaning, interpreting large datasets
- …



Can we think of more?

Acosta, J.N., Falcone, G.J., Rajpurkar, P. *et al.* Multimodal biomedical AI. *Nat Med* **28**, 1773–1784 (2022). https://doi.org/10.1038/s41591-022-01981-2

Introduction to healthcare models

Proprietary vs open source models

Proprietary
Examples: GPT-4 and -5, Med-PaLM 2,
Pros:
- State-of-the-art performance
- Strong support, frequent updates, integration
- Better safety filtering and guardrails built in

Cons:
- Costly
- Limited transparency
- Restricted fine-tuning or domain adaptation
- Data privacy concerns
- Black box

Introduction to healthcare models

Proprietary vs open source models

Proprietary
Examples: GPT-4 and -5, Med-PaLM 2,
Pros:
- State-of-the-art performance
- Strong support, frequent updates, integration
- Better safety filtering and guardrails built in

Cons:
- Costly
- Limited transparency
- Restricted fine-tuning or domain adaptation
- Data privacy concerns
- Black box

Open source
Examples: LLaMA variants, Mistral, Med-Gemma
Pros:
- Full transparency
- Can fine-tune/adapt to specific healthcare domains
- Local deployment (improved privacy)
- Community-driven innovation, ecosystem of tools

Cons:
- May lack performance
- Less robust guardrails
- Requires technical expertise
- Security and liability risks if not carefully managed
- Still a black box

Introduction to healthcare models

Proprietary vs open source models

Proprietary
Examples: GPT-4 and -5, Med-PaLM 2,
Pros:
- State-of-the-art performance
- Strong support, frequent updates, integration
- Better safety filtering and guardrails built in

Cons:
- Costly
- Limited transparency
- Restricted fine-tuning or domain adaptation
- Data privacy concerns
- Black box

Use-cases: pilots, quick prototyping, non-sensitive data tasks, high performance and when compliance can be ensured with vendor agreements.

Open source
Examples: LLaMA variants, Mistral, Med-Gemma
Pros:
- Full transparency
- Can fine-tune/adapt to specific healthcare domains
- Local deployment (improved privacy)
- Community-driven innovation, ecosystem of tools

Cons:
- May lack performance
- Less robust guardrails
- Requires technical expertise
- Security and liability risks if not carefully managed
- Still a black box

Introduction to healthcare models

Proprietary vs open source models

Proprietary
Examples: GPT-4 and -5, Med-PaLM 2,
Pros:
- State-of-the-art performance
- Strong support, frequent updates, integration
- Better safety filtering and guardrails built in

Cons:
- Costly
- Limited transparency
- Restricted fine-tuning or domain adaptation
- Data privacy concerns
- Black box

Use-cases: pilots, quick prototyping, non-sensitive data tasks, high performance and when compliance can be ensured with vendor agreements.

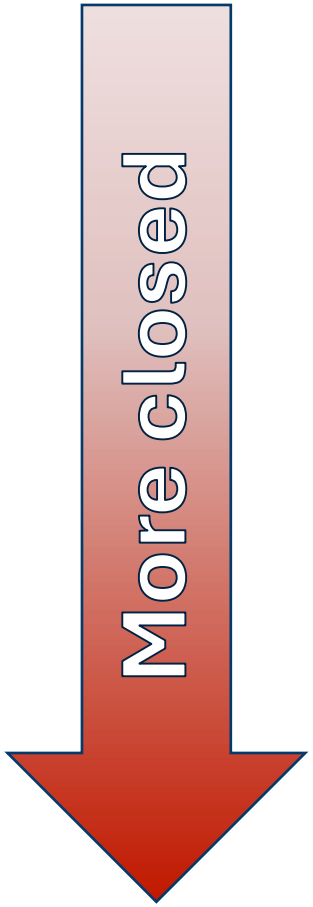
Open source
Examples: LLaMA variants, Mistral, Med-Gemma
Pros:
- Full transparency
- Can fine-tune/adapt to specific healthcare domains
- Local deployment (improved privacy)
- Community-driven innovation, ecosystem of tools

Cons:
- May lack performance
- Less robust guardrails
- Requires technical expertise
- Security and liability risks if not carefully managed
- Still a black box

Use-cases: research, long-term institutional control, tasks involving sensitive data, and when transparency & explainability are priorities.

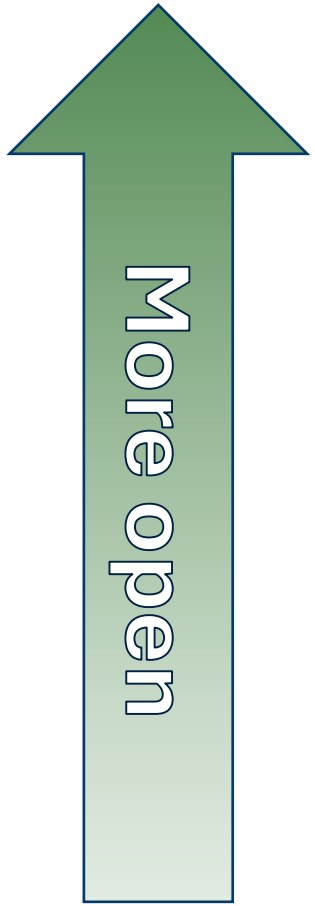
Introduction to healthcare models

What is open source?



1. Fully Open (Weights, Code, and Data) (Pythia, RedPajama)

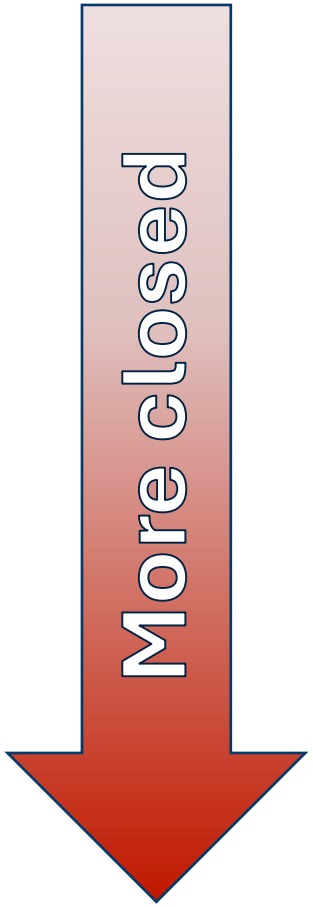
- **What's open?** Architecture, weights, training pipeline, dataset
- **Implication:** Maximum transparency, enables reproducibility and auditing



UMC Utrecht

Introduction to healthcare models

What is open source?

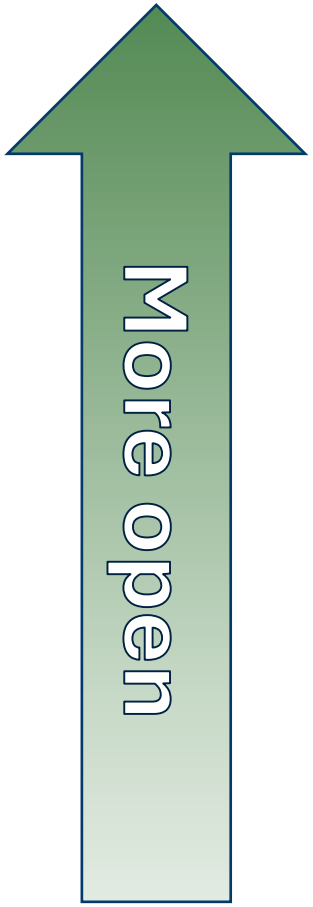


1. Fully Open (Weights, Code, and Data) (Pythia, RedPajama)

- **What's open?** Architecture, weights, training pipeline, dataset
- **Implication:** Maximum transparency, enables reproducibility and auditing

2. Open-Weight + Limited Data Transparency (Falcon, BioMistral)

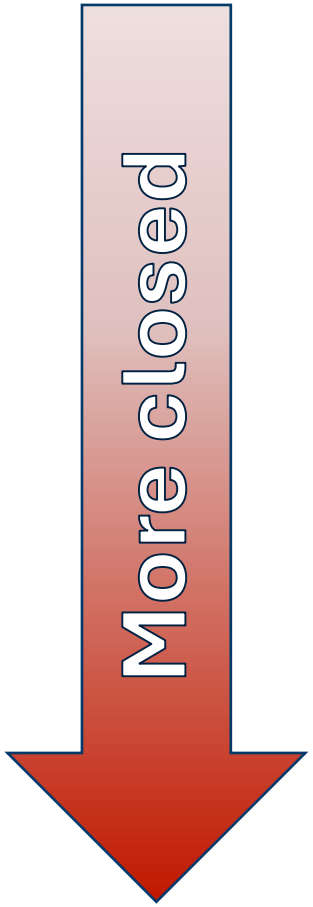
- **What's open?** Weights + partial data documentation
- **Closed:** Full dataset not accessible
- **Implication:** Better understanding of biases, still not fully auditable



UMC Utrecht

Introduction to healthcare models

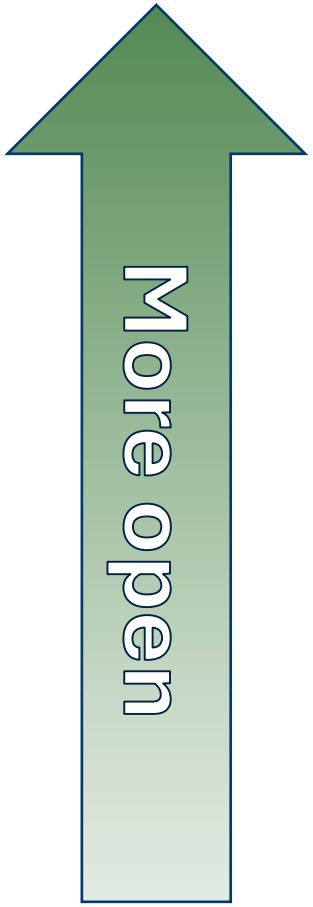
What is open source?



1. Fully Open (Weights, Code, and Data) (Pythia, RedPajama)
- **What's open?** Architecture, weights, training pipeline, dataset
- **Implication:** Maximum transparency, enables reproducibility and auditing

2. Open-Weight + Limited Data Transparency (Falcon, BioMistral)
- **What's open?** Weights + partial data documentation
- **Closed:** Full dataset not accessible
- **Implication:** Better understanding of biases, still not fully auditable

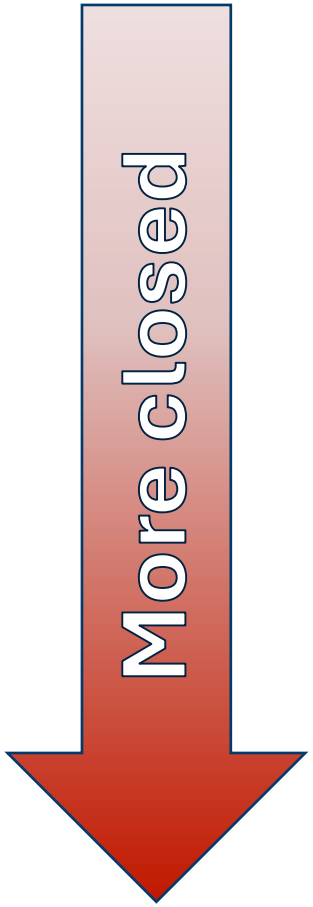
3. Semi-Open (Architecture Open, Data Closed) (LLaMA 3, Med-Gemma)
- **What's open?** Model architecture + weights
- **Closed:** Training data (sources unknown or partially disclosed)
- **Implication:** Reproducible inference & fine-tuning possible, but may contain hidden biases



UMC Utrecht

Introduction to healthcare models

What is open source?

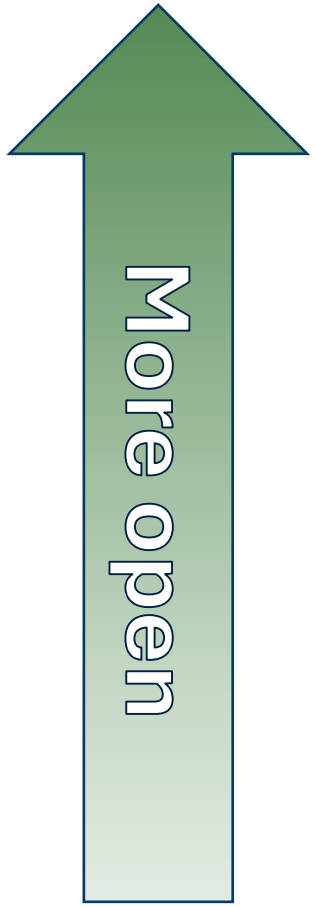


1. Fully Open (Weights, Code, and Data) (Pythia, RedPajama)
- **What's open?** Architecture, weights, training pipeline, dataset
- **Implication:** Maximum transparency, enables reproducibility and auditing

2. Open-Weight + Limited Data Transparency (Falcon, BioMistral)
- **What's open?** Weights + partial data documentation
- **Closed:** Full dataset not accessible
- **Implication:** Better understanding of biases, still not fully auditable

3. Semi-Open (Architecture Open, Data Closed) (LLaMA 3, Med-Gemma)
- **What's open?** Model architecture + weights
- **Closed:** Training data (sources unknown or partially disclosed)
- **Implication:** Reproducible inference & fine-tuning possible, but may contain hidden biases

4. Fully Proprietary (GPT-4, Claude 3)
- **What's open?** Only the API access
- **Closed:** Architecture, weights, training data
- **Implication:** High performance, but black box; limited trust/explainability



UMC Utrecht

Introduction to healthcare models

Size matters

Small Models (1-10B parameters)

- Can run on (good) laptop (8-16GB VRAM)
- Memory footprint: 4-20GB
- Edge devices with optimizations (quantization, pruning)
- Cost efficiency: $0.05-0.15/hour on cloud services

Introduction to healthcare models

Size matters

Small Models (1-10B parameters)

- Can run on (good) laptop (8-16GB VRAM)
- Memory footprint: 4-20GB
- Edge devices with optimizations (quantization, pruning)
- Cost efficiency: $0.05-0.15/hour on cloud services

Medium Models (10-70B parameters)

- Workstation-class GPUs necessary (24-100GB VRAM)
- Single high-end GPU (A10, RTX 4090, A100) with quantization
- Multi-GPU setups for full precision
- Cloud-based deployment with mid-tier instances
- Cost efficiency: $0.20-1.00/hour on cloud services

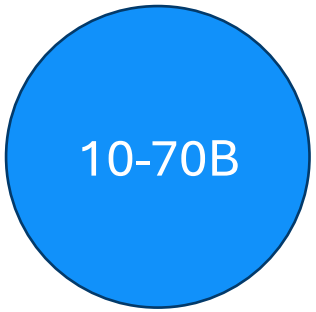
Introduction to healthcare models

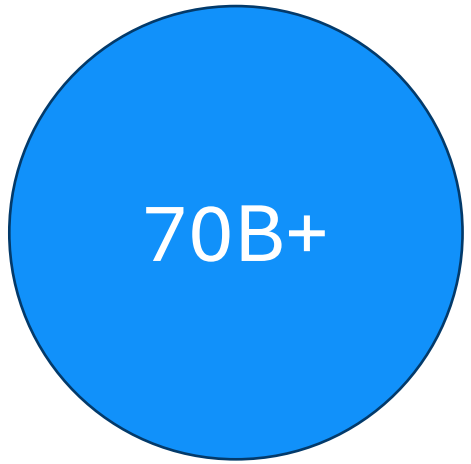
Size matters

Small Models (1-10B parameters)
- Can run on (good) laptop (8-16GB VRAM)
- Memory footprint: 4-20GB
- Edge devices with optimizations (quantization, pruning)
- Cost efficiency: $0.05-0.15/hour on cloud services

Medium Models (10-70B parameters)
- Workstation-class GPUs necessary (24-100GB VRAM)
- Single high-end GPU (A10, RTX 4090, A100) with quantization
- Multi-GPU setups for full precision
- Cloud-based deployment with mid-tier instances
- Cost efficiency: $0.20-1.00/hour on cloud services

Large Models (70B+ parameters)
- Data center GPUs or specialized AI accelerators (80GB+ VRAM)
- Multiple high-end GPUs (A100, H100) in parallel
- Distributed computing across multiple machines
- Specialized AI cloud services with optimized infrastructure
- Cost efficiency: $1.50-10.00+/hour on cloud services





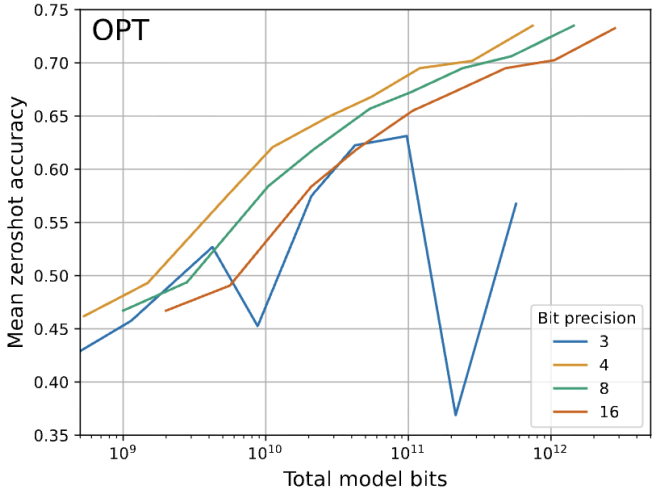
UMC Utrecht

Introduction to healthcare models

Quantization

Concept:
Reduce precision of model weights and activations.
Convert FP32 to FP16, INT8, or INT4.

Benefits:
Cuts memory use and boosts speed.
Enables use on laptop or smaller devices.
Minimal accuracy loss for most quantizations.

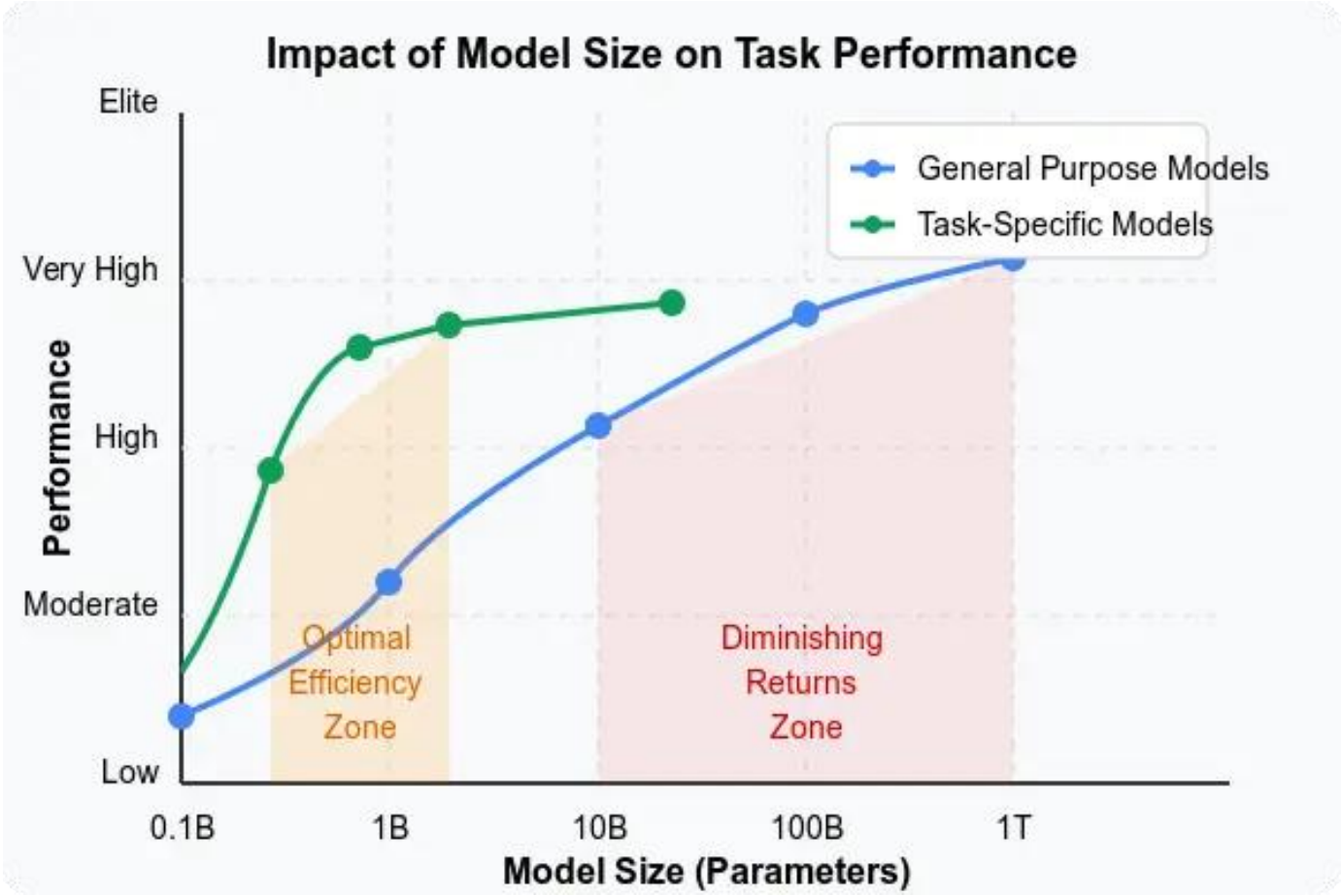


Precision	Memory	Speed	Accuracy Drop
FP32	100%	1×	0%
FP16	~50%	1.5–2×	<1%
INT8	~25%	2–4×	1–5%
INT4	~12%	4–8×	5–8%

UMC Utrecht

Introduction to healthcare models

But it's more about how you use it!



https://www.analyticsvidhya.com/blog/2025/03/human-side-of-llm-model-sizes/

Medical models

Dedicated models vs. general LLMs

General LLMs

Pros:

- Huge training data -> broad knowledge, flexible reasoning.
- Strong at general tasks.
- More mature ecosystem and availability

Dedicated models

Pros:

- Trained on domain-specific data.
- Stronger medical vocabulary & reasoning.
- Better alignment with clinical tasks.

Medical models

Dedicated models vs. general LLMs

General LLMs

Pros:

- Huge training data -> broad knowledge, flexible reasoning.
- Strong at general tasks.
- More mature ecosystem and availability

Cons:

- Lack specialized clinical knowledge.
- May hallucinate or misuse medical terminology.
- Risk of unsafe outputs.
- (Legally) not allowed for healthcare purposes!

Dedicated models

Pros:

- Trained on domain-specific data.
- Stronger medical vocabulary & reasoning.
- Better alignment with clinical tasks.

Cons:

- Narrower knowledge base.
- Smaller training sets -> risk of bias, overfitting.
- Less available/commercial support than general LLMs.
- Weaker in emergent capabilities (understanding, reasoning)

LLM validation

With a focus on healthcare

The importance of (clinical) validation

- Clinical settings require accuracy, safety, and fairness.
- Incorrect output has high risk (risk = probability x severity)
- Validation should cover more than only performance.
- Human oversight remains necessary!

Must assess:

- Linguistic quality (does it make sense grammatically?)
- (Semantic) correctness (does it makes sense?)
- Clinical validity (is it clinically correct/beneficial?)
- Clinical utility (does it actually help?)

LLM validation

With a focus on healthcare

The importance of (clinical) validation

- Clinical settings require accuracy, safety, and fairness.
- Incorrect output has high risk (risk = probability x severity)
- Validation should cover more than only performance.
- Human oversight remains necessary!

Must assess:

- Linguistic quality (does it make sense grammatically?)
- (Semantic) correctness (does it makes sense?)
- Clinical validity (is it clinically correct/beneficial?)
- Clinical utility (does it actually help?)

We will discuss:

Perplexity, lexical similarity, semantic similarity, LLM-as-a-judge, human evaluation

LLM validation

With a focus on healthcare

The importance of (clinical) validation

- Clinical settings require accuracy, safety, and fairness.
- Incorrect output has high risk (risk = probability x severity)
- Validation should cover more than only performance.
- Human oversight remains necessary!

Must assess:

• Linguistic quality (does it make sense grammatically?)	Perplexity
• (Semantic) correctness (does it makes sense?)	Lexical similarity, semantic similarity
• Clinical validity (is it clinically correct/beneficial?)	LLM-as-a-judge, human evaluation
• Clinical utility (does it actually help?)	Human evaluation

We will discuss:

Perplexity, lexical similarity, semantic similarity, LLM-as-a-judge, human evaluation

LLM validation: fluency

The perplexity metric

Perplexity

- Meaning: confused or surprised
- Measures fluency
- A metric of how **perplexed** a model is when presented with the actual correct next word
- Lower is better
- **Perplexity is the exponentiated cross-entropy between the model's predicted token distribution and the true next tokens in the reference text.**

LLM validation: fluency

The perplexity metric

Perplexity

- Meaning: confused or surprised
- Measures fluency
- A metric of how **perplexed** a model is when presented with the actual correct next word
- Lower is better
- **Perplexity is the exponentiated cross-entropy between the model's predicted token distribution and the true next tokens in the reference text.**

For a test set T containing N tokens, and a model m that assigns probability $m(T)$ to that text, perplexity is define

$$\mathrm{PPL}(T) = \left(\frac{1}{\prod_{i=1}^{N} P(t_i)}\right)^{1/N} = \left(\frac{1}{m(T)}\right)^{1/N} = 2^{-\frac{1}{N}\log_2 m(T)} = 2^{H(p;m)}$$

LLM validation: fluency

The perplexity metric - example

$$\mathrm{PPL}(T) = \left(\frac{1}{m(T)}\right)^{1/N}$$

Reference text: “The patient recovered”

Predicted probability of this sequence is:

Token	Model probability
"The"	0.2
"patient"	0.1
"recovered"	0.5

$m(T) = 0.2 \times 0.1 \times 0.5 = 0.01$

Number of tokens: $N = 3$

$PPL(T) = (1 / m(T))^{1/N} = (100)^{1/3} \approx 4.64$

So, the **perplexity = 4.6**

This can be interpreted as meaning the model is as uncertain as if it had to choose between roughly **5 equally likely options** for each token.

LLM validation: lexical similarity

BLUE, ROUGE

ROUGE(-N):

- How many n-grams of the generated text are also present in the references, divided by the number n-grams in the reference text
- **Measure of recall:**

$$\text{ROUGE-N} = \frac{\#\text{ of overlapping n-grams}}{\#\text{ of n-grams in reference}}$$

LLM validation: lexical similarity

BLUE, ROUGE

ROUGE(-N):

- How many n-grams of the generated text are also present in the references, divided by the number n-grams in the reference text
- **Measure of recall:**

$$\text{ROUGE-N} = \frac{\#\text{ of overlapping n-grams}}{\#\text{ of n-grams in reference}}$$

BLEU:

- How many n-grams of the generated text are also present in the reference, divided by the number of n-grams in the generated text
- **Measure of precision:**

$$\text{BLEU} = \text{BP} \times \left(\prod_{n=1}^{N} p_n^{w_n} \right)$$

- p_n = proportion of overlapping n-grams of size *n*
- w_n = weight for each n-gram (usually equal, e.g. ¼ each for up to 4-grams)
- BP = brevity penalty

UMC Utrecht

*Brevity penalty is added to avoid getting high scores for very short answers that include some words that are in the reference

LLM validation: lexical similarity

BLUE, ROUGE

ROUGE(-N):

- How many n-grams of the generated text are also present in the references, divided by the number n-grams in the reference text
- **Measure of recall:**

$$\text{ROUGE-N} = \frac{\#\text{ of overlapping n-grams}}{\#\text{ of n-grams in reference}}$$

BLEU:

- How many n-grams of the generated text are also present in the reference, divided by the number of n-grams in the generated text
- **Measure of precision:**

$$\text{BLEU} = \text{BP} \times \left(\prod_{n=1}^{N} p_n^{w_n} \right)$$

$$p_1 = \frac{\#\text{ of overlapping 1-grams}}{\#\text{ of 1-grams in candidate}}$$

$$p_2 = \frac{\#\text{ of overlapping 2-grams}}{\#\text{ of 2-grams in candidate}}$$

$$p_3 = \frac{\#\text{ of overlapping 3-grams}}{\#\text{ of 3-grams in candidate}}$$

$$p_4 = \frac{\#\text{ of overlapping 4-grams}}{\#\text{ of 4-grams in candidate}}$$

$$\text{BLEU} = \text{BP} \times (p_1 \times p_2 \times p_3 \times p_4)^{1/4}$$

Geometric mean

UMC Utrecht

*Brevity penalty is added to avoid getting high scores for very short answers that include some words that are in the reference

LLM validation: semantic similarity

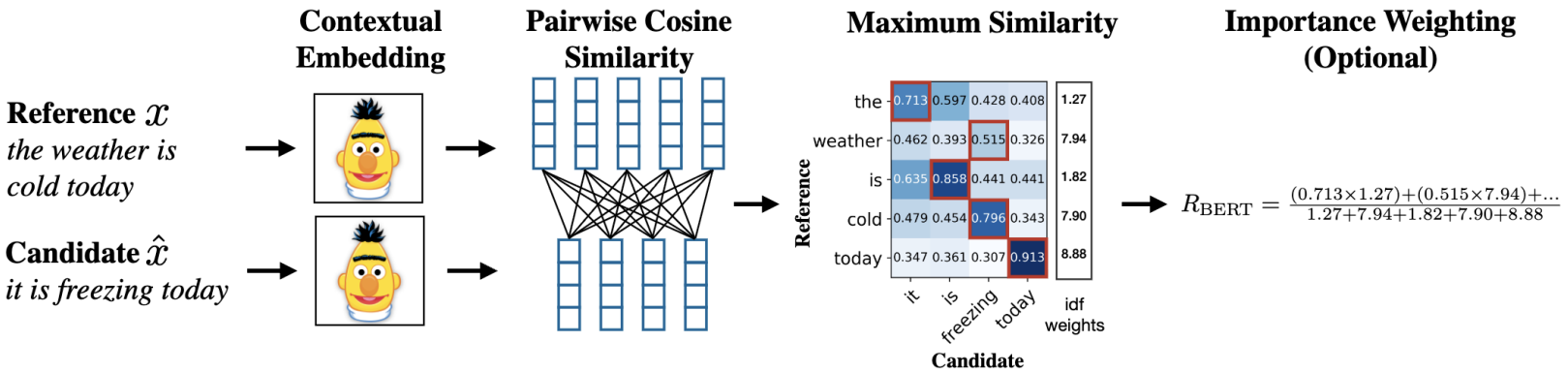
Using embeddings to test similarity

Concept:

- Measures similarity in meaning, not wording.
- Takes context into account.
- Uses pretrained embeddings (e.g. BERT)

How it works:

- Encode candidate and reference sentences.
- Compute cosine similarity between embeddings.
- Match tokens to most similar counterpart.
- Average similarities to get the score.



LLM validation: semantic similarity

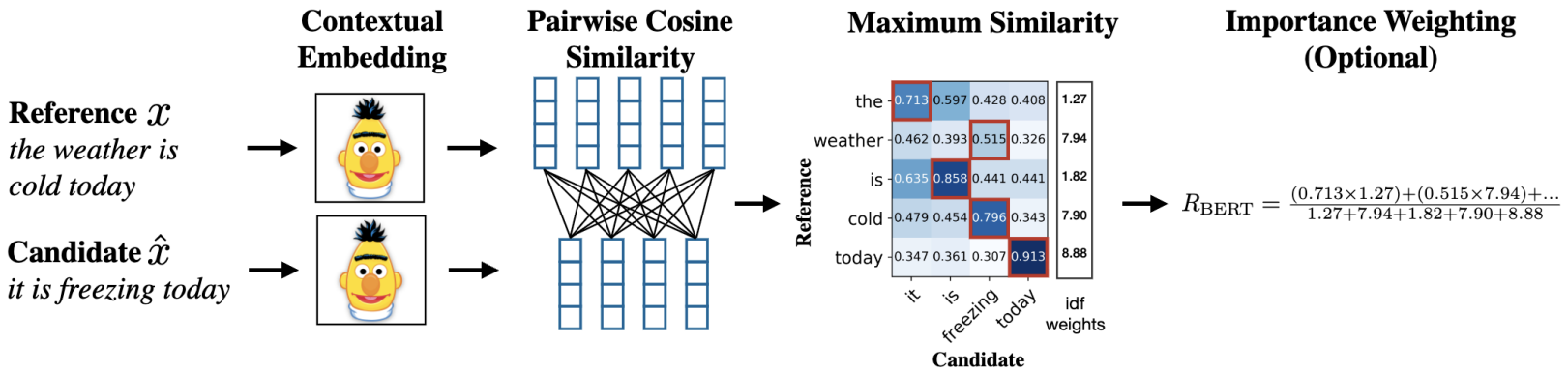
Using embeddings to test similarity

Concept:
- Measures similarity in meaning, not wording.
- Takes context into account.
- Uses pretrained embeddings (e.g. BERT)

How it works:
- Encode candidate and reference sentences.
- Compute cosine similarity between embeddings.
- Match tokens to most similar counterpart.
- Average similarities to get the score.

Advantages:
- Captures meaning and paraphrases.
- Insensitive to word order.
- Correlates better with human judgement.

Limitations:
- Requires sufficiently powerful model.
- Less interpretable than lexical overlap metrics.



UMC Utrecht

LLM validation: LLM-as-a-judge

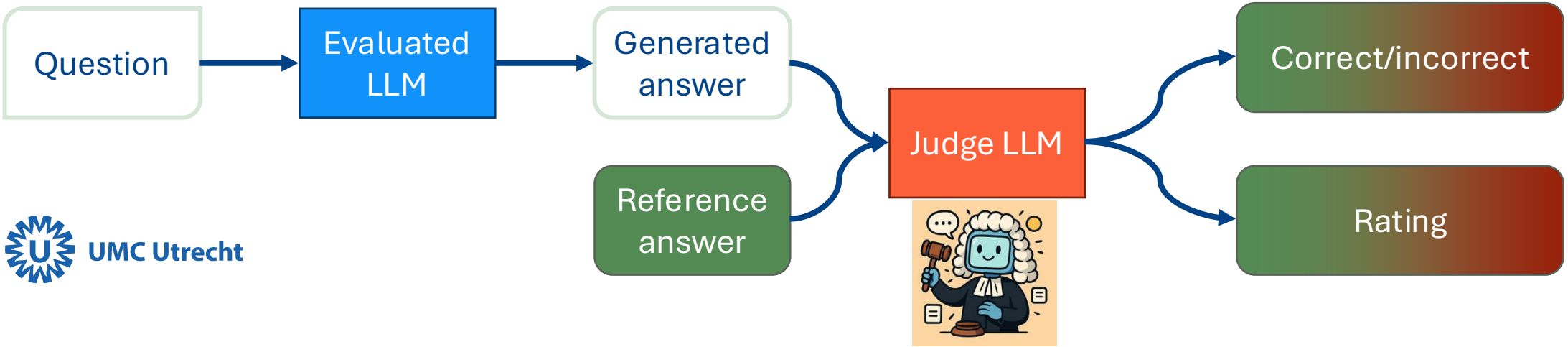
Larger models checks smaller model

Concept:

- Use LLMs to judge generated text quality.

How it works:

- Give task, reference, and model answers.
- Ask LLM which response matches reference.
- LLM rates fluency, accuracy, and reasoning.



UMC Utrecht

LLM validation: LLM-as-a-judge

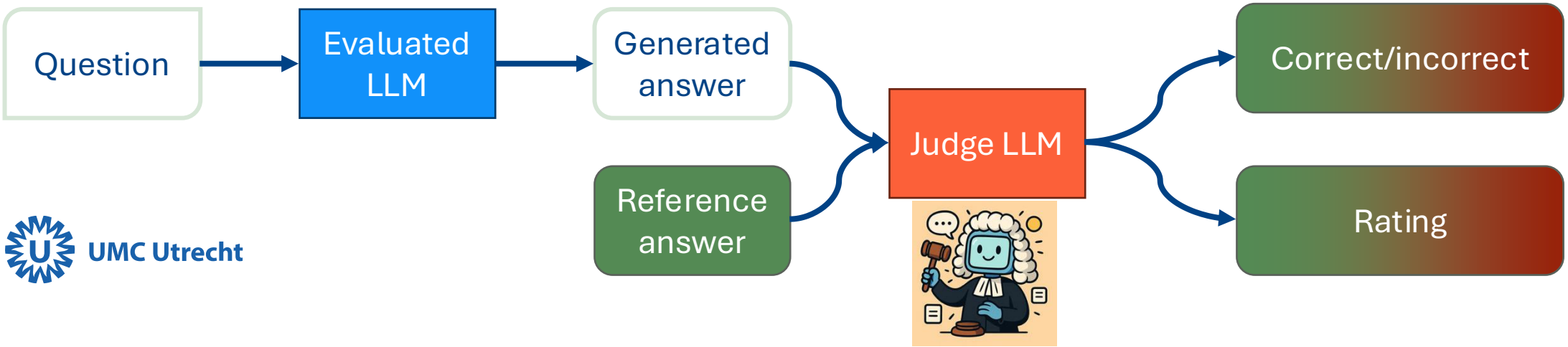
Larger models checks smaller model

Concept:
- Use LLMs to judge generated text quality.

How it works:
- Give task, reference, and model answers.
- Ask LLM which response matches reference.
- LLM rates fluency, accuracy, and reasoning.

Advantages:
- Understands meaning, not just word overlap.
- Correlates strongly with human judgments.
- Also works even without reference answers.

Limitations:
- Can favour verbose or stylish outputs.
- Results depend on prompt and model quality.
- Needs checks for fairness and bias.



UMC Utrecht

Model and data sources

Some examples

Kaggle: Public competitions and datasets, often imaging-focused, easy to access. Also hosts models.
Examples:

- RSNA Pneumonia Detection Challenge (X-rays).
- SIIM-ISIC Melanoma classification (skin lesions).
- COVID-19 Open Research Dataset Challenge (research text)

Hugging Face Hub: Repository with both models and datasets, and also responsible for some of the most used python libraries. Also able to host your model online using 'Spaces'.
Examples:

- PubMedQA (biomedical QA).
- RadGraph (radiology report entities/relations).

PhysioNet: Collection of physiological signal datasets, often time-series.
Examples:

- MIT-BIH Arrhythmia Database (ECG).
- Sleep-EDF Database (EEG, sleep studies).

MIMIC-IV (available through PhysioNet): Large, de-identified ICU dataset (structured + unstructured EHR, imaging links). Requires credentialed access.
Examples:

- ICU patient vitals, labs, medications.
- Linked imaging studies (MIMIC-CXR).

Practical session 1

Fine-tuning a pre-trained model for healthcare

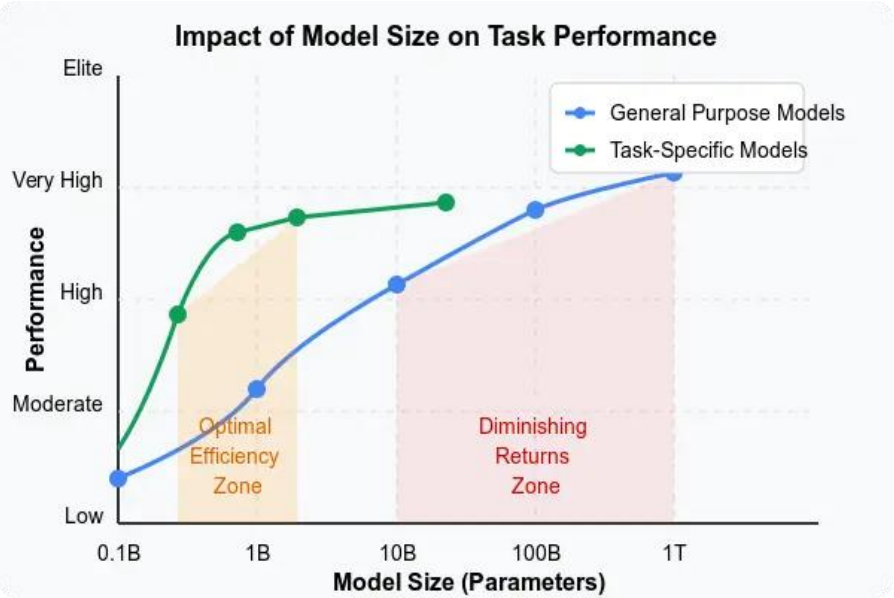
General models perform quite well on all tasks

Specialized models perform better on a single task

Specialized models can be smaller with similar performance to larger general models

Let's test this hypothesis!

LINK TO COLAB NOTEBOOK



Short break

Multimodal model architecture

Architecture basics

To be able to process more than just text data, we need to combine the data somewhere in our network

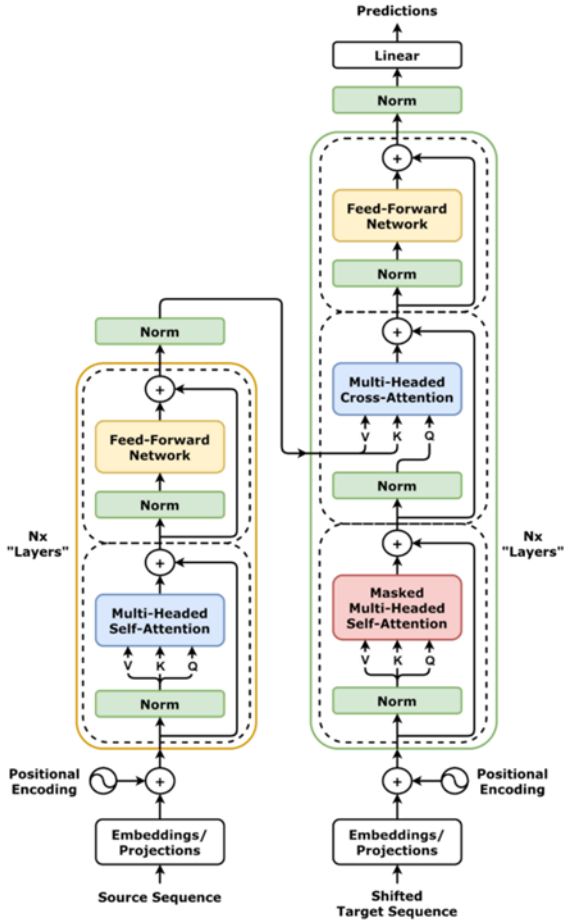
Three main components

Encoder network: Specialized for each modality, embedding input

- **Text:** Transformer encoder (BERT, LLaMA).
- **Images:** CNNs, Vision Transformers.
- **Audio/signals:** 1D CNNs, wav2vec, RNNs.

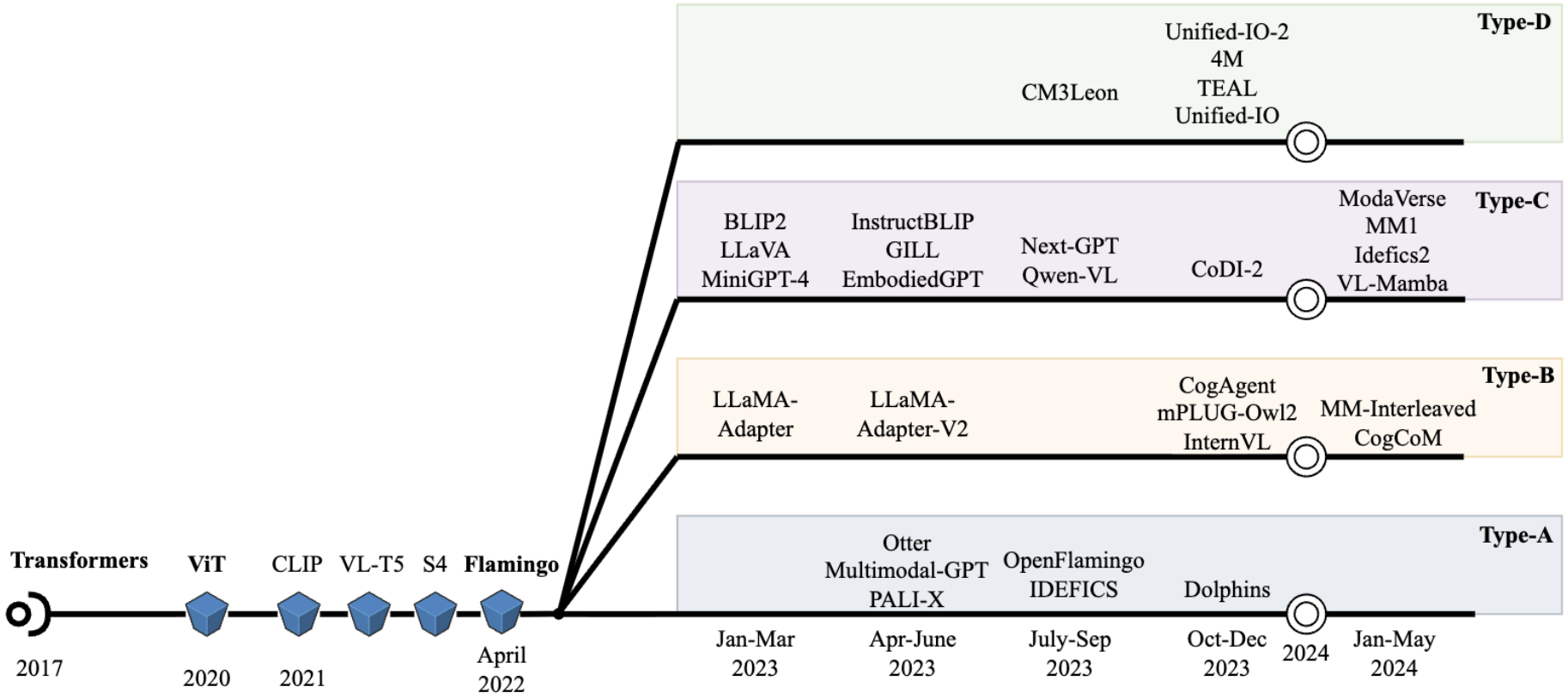
Fusion layer: Combines encoded features
(for example through concatenation or cross-attention).

Decoder / head: Generates text, classification labels, images, segmentations, sound or sequences, depending on output modality.



Multimodal model architecture

Architecture basics

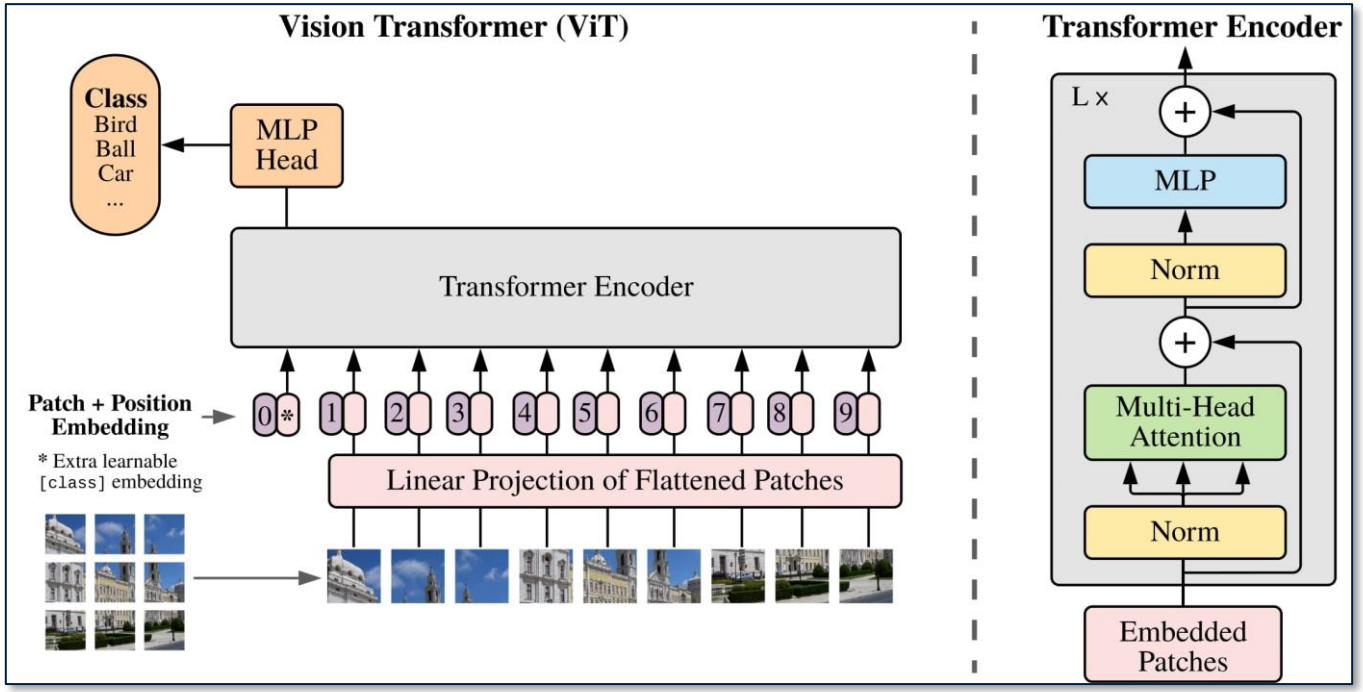


Vision-language models

How to ingest images into a transformer model

To embed images in similar way as words, we need to get them into an embedding vector

- For an image with shape H x W (height x width)
- Choose patch size P x P
- The number of patches from an image is: N=(H x W)/P^2
- Each patch is flattened and becomes an embedding vector



UMC Utrecht

Vision-language models

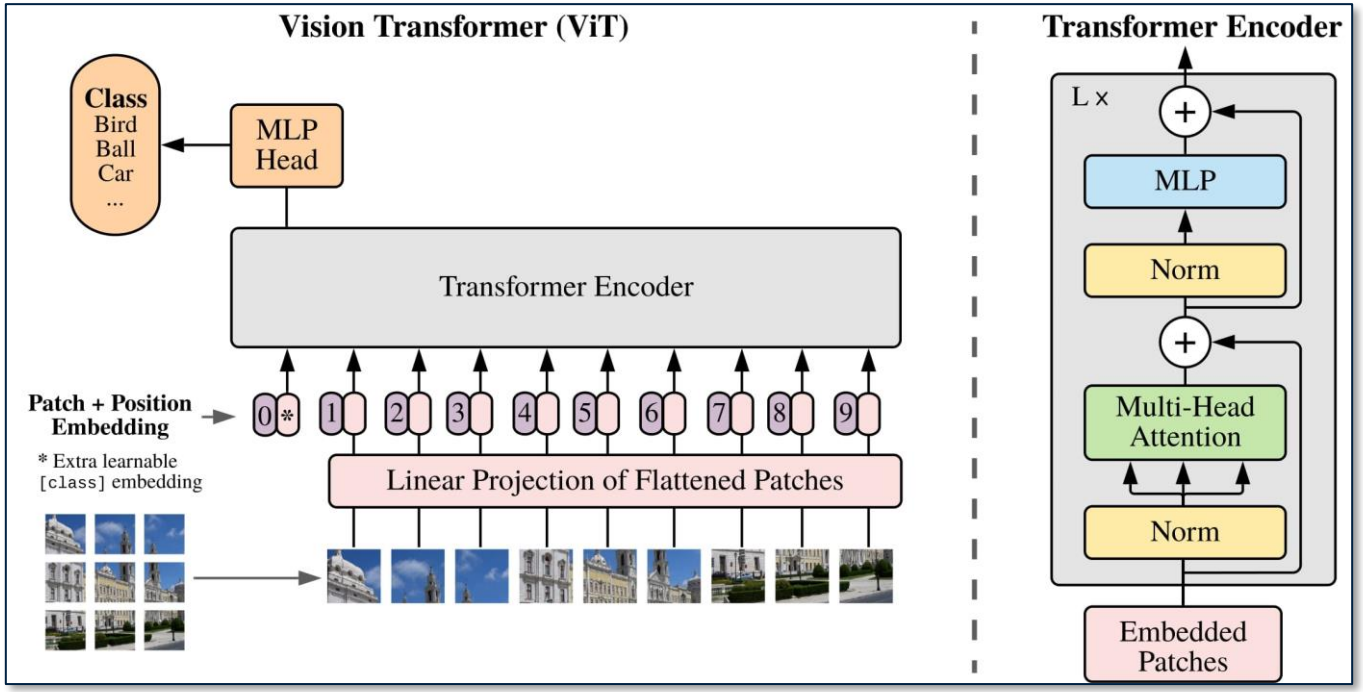
How to ingest images into a transformer model

To embed images in similar way as words, we need to get them into an embedding vector

- For an image with shape H x W (height x width)
- Choose patch size P x P
- The number of patches from an image is: N=(H x W)/P^2
- Each patch is flattened and becomes an embedding vector

Example

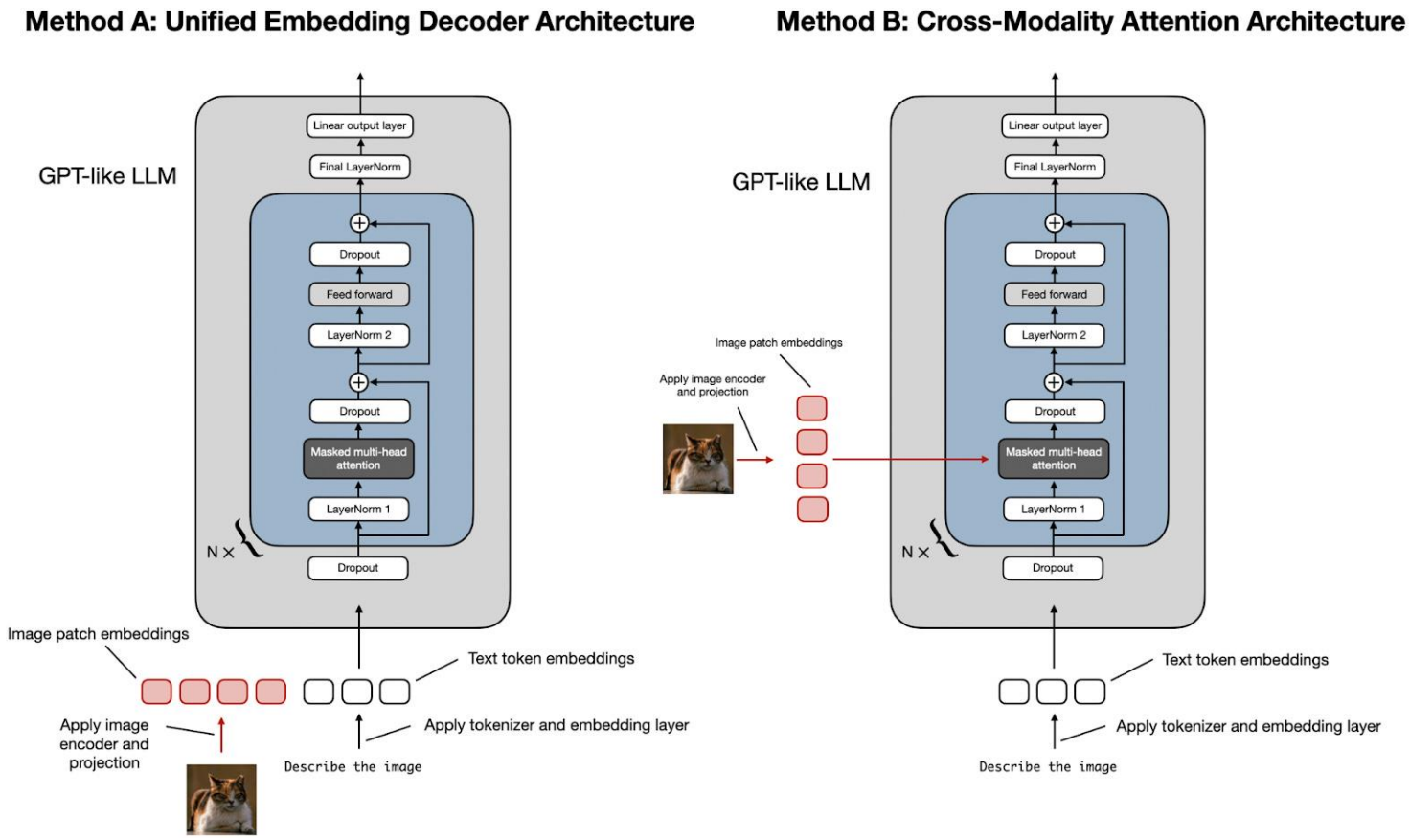
- 224×224RGB image with 16×16patches
- $224^2/16^2$ =196 tokens
- Each patch is flattened
- Linearly projected into a embedding vector



UMC Utrecht

Multimodal model architecture

Two main approaches

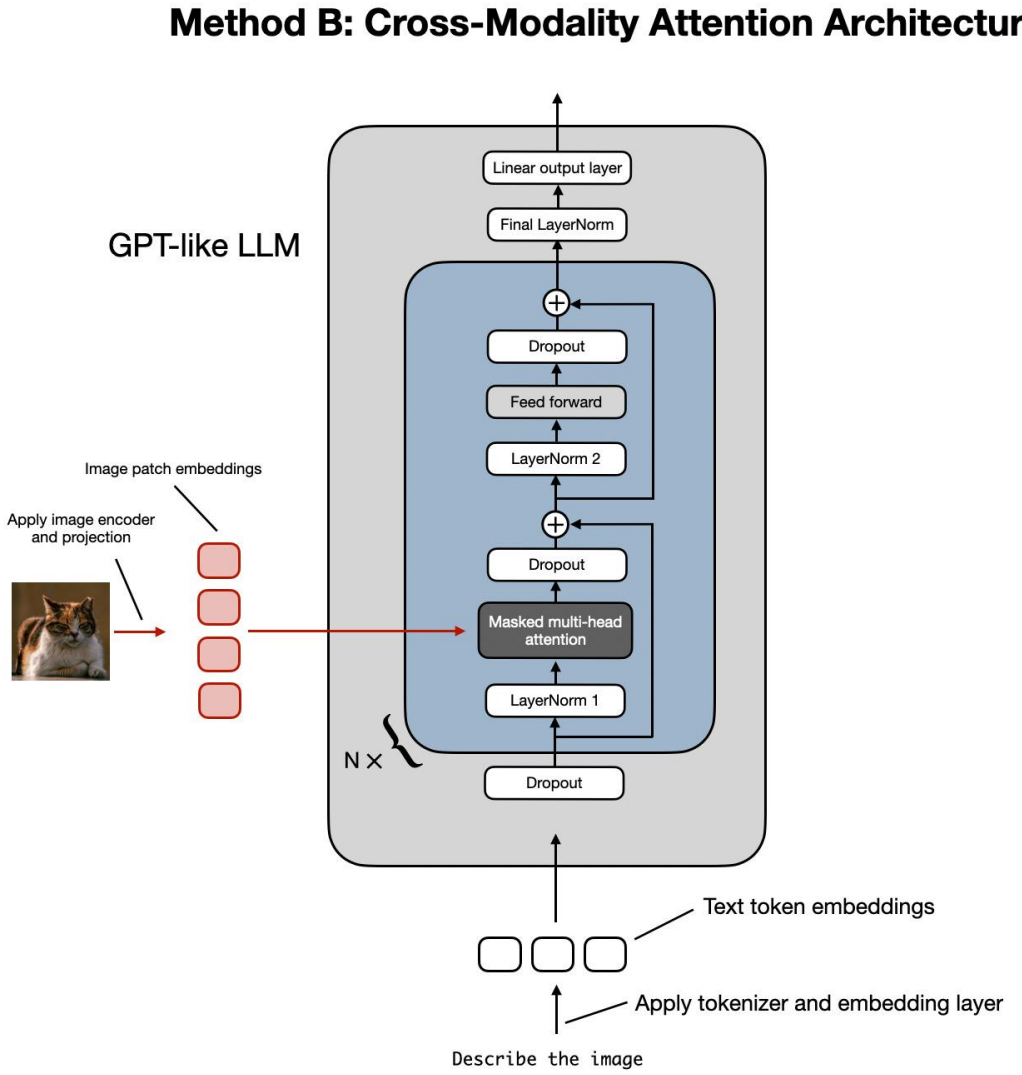


Source: https://magazine.sebastianraschka.com/p/understanding-multimodal-llms

Multimodal model architecture

Cross-modality attention

- Separate encoders for image and text
- Fused via cross-attention in transformer layers
- Maintains modality-specific representations
- Either one or both of the encoders often pre-trained
- Either one of the encoders often frozen
- Enables richer multimodal interactions



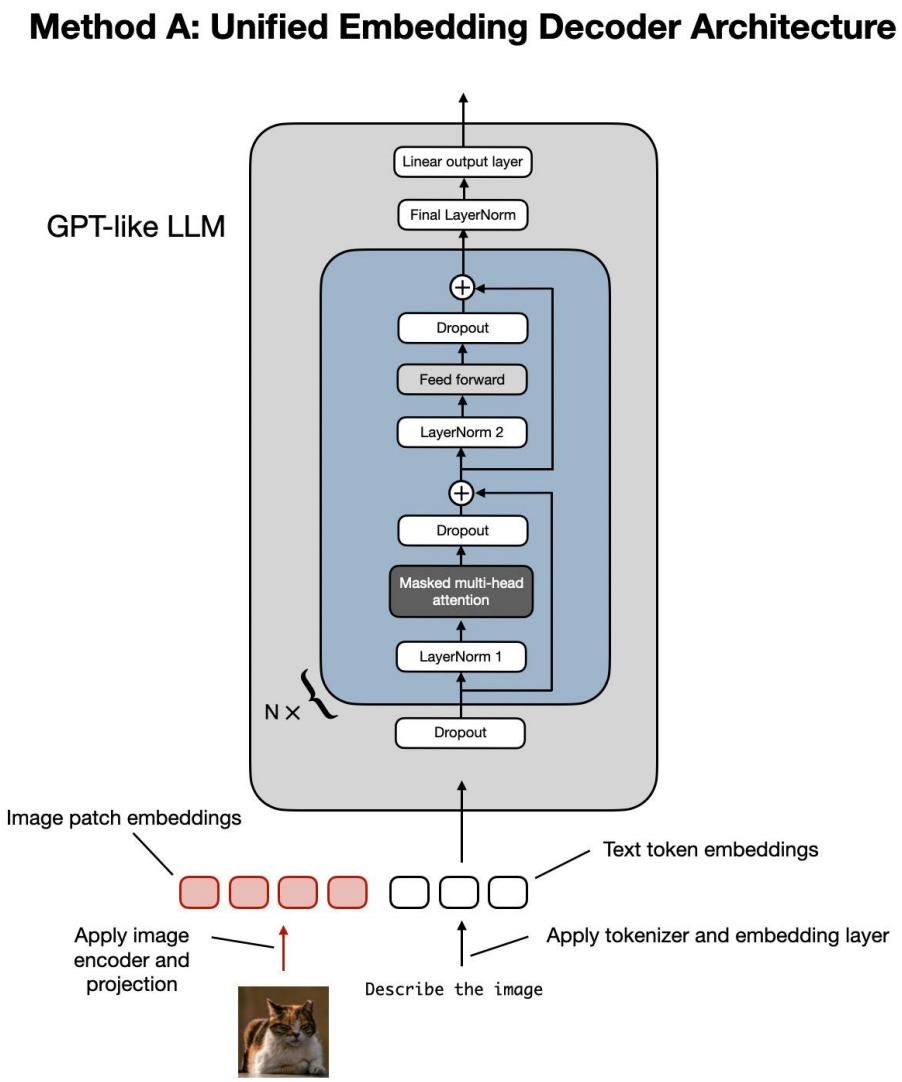
Source: https://magazine.sebastianraschka.com/p/understanding-multimodal-llms

Multimodal model architecture

Unified embedding decoder

Unified embedding:

- One shared transformer decoder for all inputs
- Image patches projected to text-sized tokens
- Tokens from all modalities concatenated
- No architectural change to base LLM
- Simple but limited multimodal reasoning



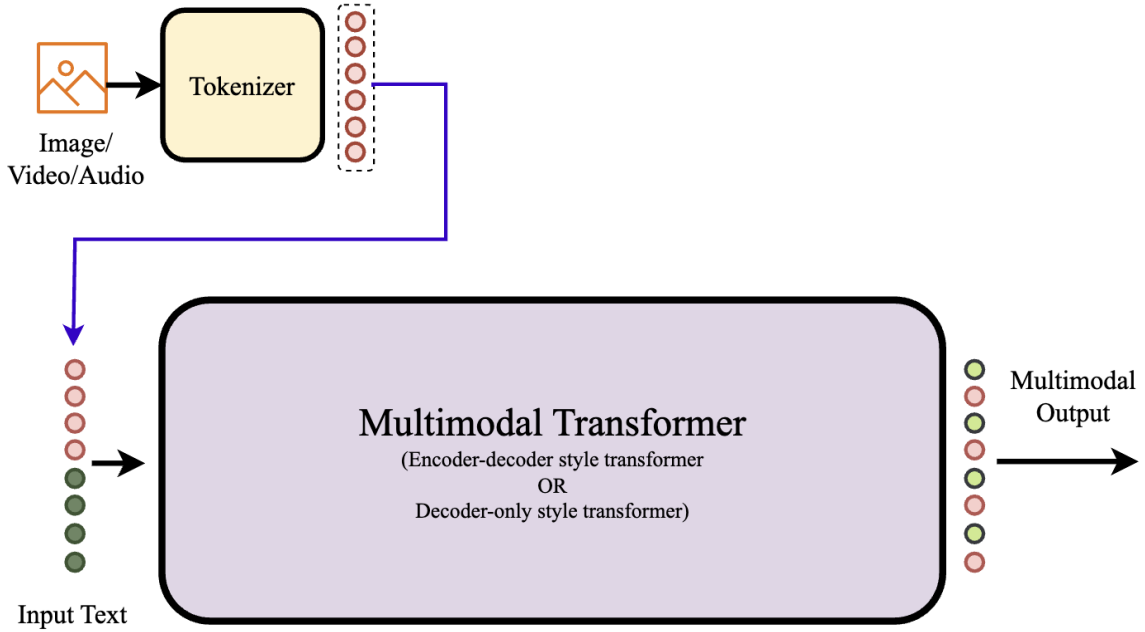
Source: https://magazine.sebastianraschka.com/p/understanding-multimodal-llms

Multimodal model architecture

Unified embedding decoder

Unified embedding:

- One shared transformer decoder for all inputs
- Image patches projected to text-sized tokens
- Tokens from all modalities concatenated
- No architectural change to base LLM
- Simple but limited multimodal reasoning



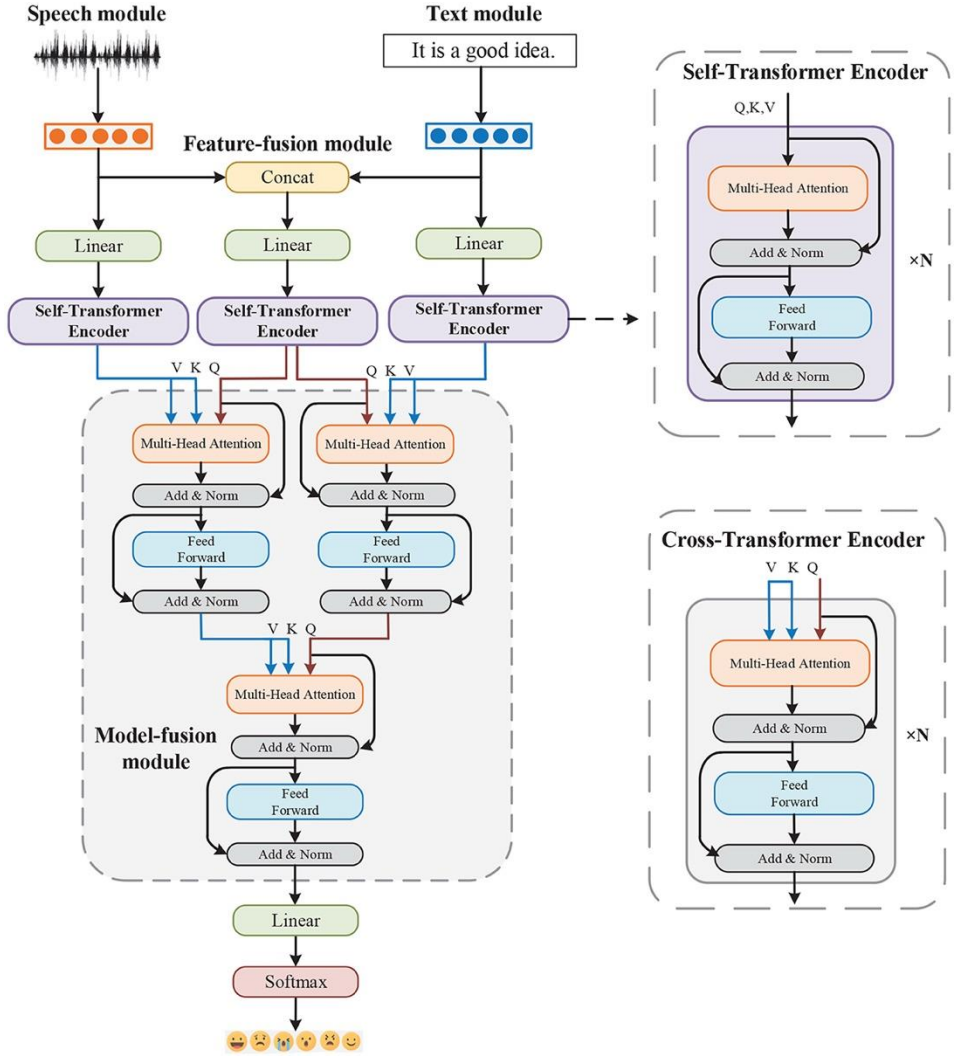
Source: https://arxiv.org/pdf/2405.17927

Multimodal model architecture examples

Example architectures: Speech emotion detection

Multimodal Transformer Fusion (Wang et al., 2023)

- Combines speech and text for emotion recognition
- Uses self- and cross-transformers for fusion
- Outperforms single-modality models



Source: Wang, Yuanyuan, et al. Multimodal transformer augmented fusion for speech emotion recognition. *Frontiers in neurorobotics*, 2023, 17: 1181598.

Multimodal model architecture examples

Palm-E

Palm-E: Embodied Multimodal Language Model

- Builds on PaLM; injects vision & state inputs into LLM.
- Inputs: text + images + robot sensor state
- Same embedding space: images encoded like word tokens.
- Handles robot planning, VQA, captioning tasks
- Largest variant: ~562 billion parameters

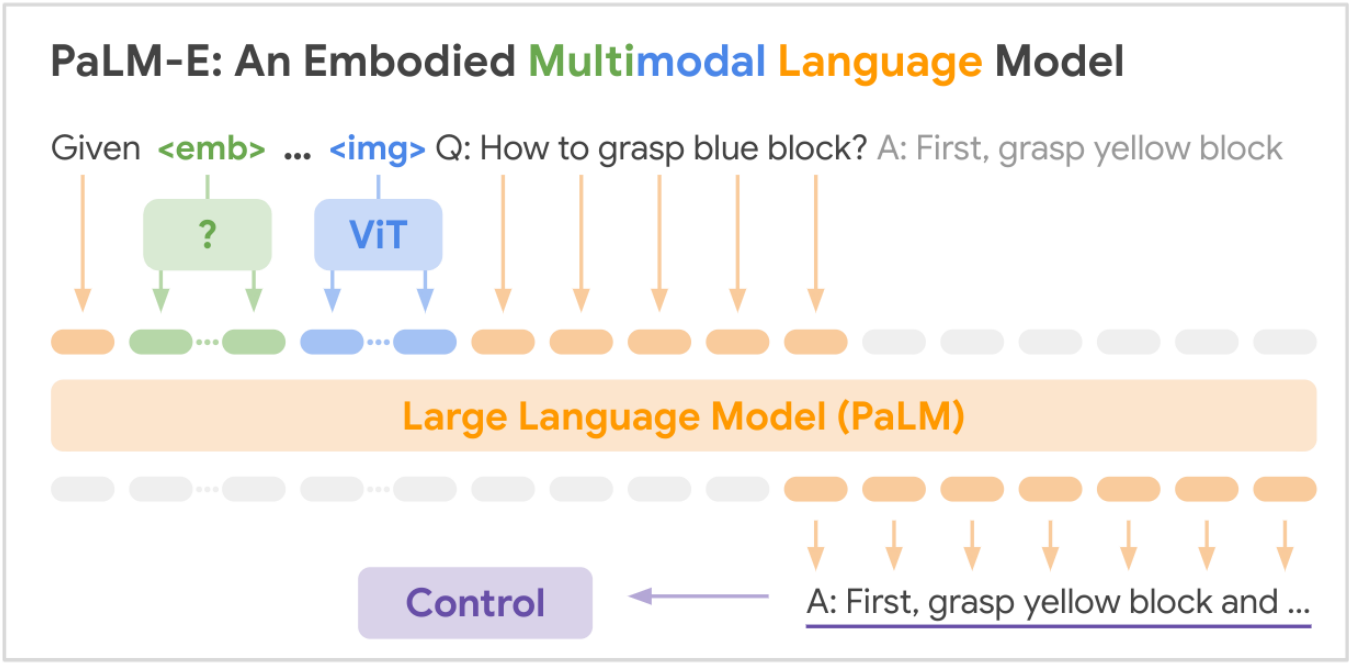
Source: https://arxiv.org/pdf/2303.03378

Multimodal model architecture examples

Palm-E

Palm-E: Embodied Multimodal Language Model

- Builds on PaLM; injects vision & state inputs into LLM.
- Inputs: text + images + robot sensor state
- Same embedding space: images encoded like word tokens.
- Handles robot planning, VQA, captioning tasks
- Largest variant: ~562 billion parameters



Source: https://arxiv.org/pdf/2303.03378

Multimodal model architecture examples

Multimodal Llama 3

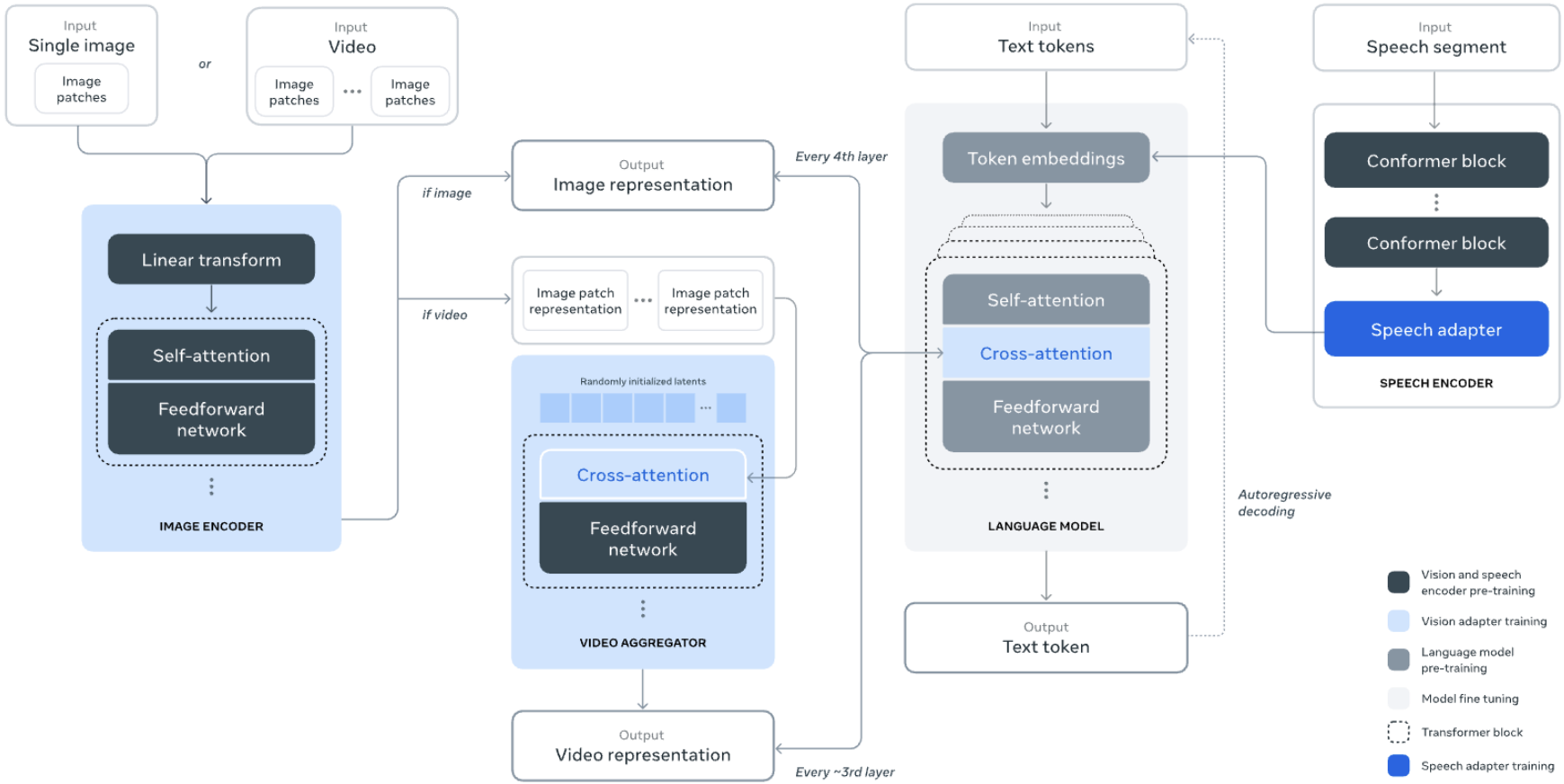


Figure 28 Illustration of the compositional approach to adding multimodal capabilities to Llama 3 that we study in this paper. This approach leads to a multimodal model that is trained in five stages: **(1)** language model pre-training, **(2)** multi-modal encoder pre-training, **(3)** vision adapter training, **(4)** model finetuning, and **(5)** speech adapter training.

Examples

Google MedGemma

What is Med-Gemma?

- Google's open multimodal models for healthcare.
- Focused on radiology, clinical reasoning, and multimodal decision support.
- Based on Gemma 3.
- Sizes: 4B and 27B (allows for local experimentation)
- Recent release (July 2025)

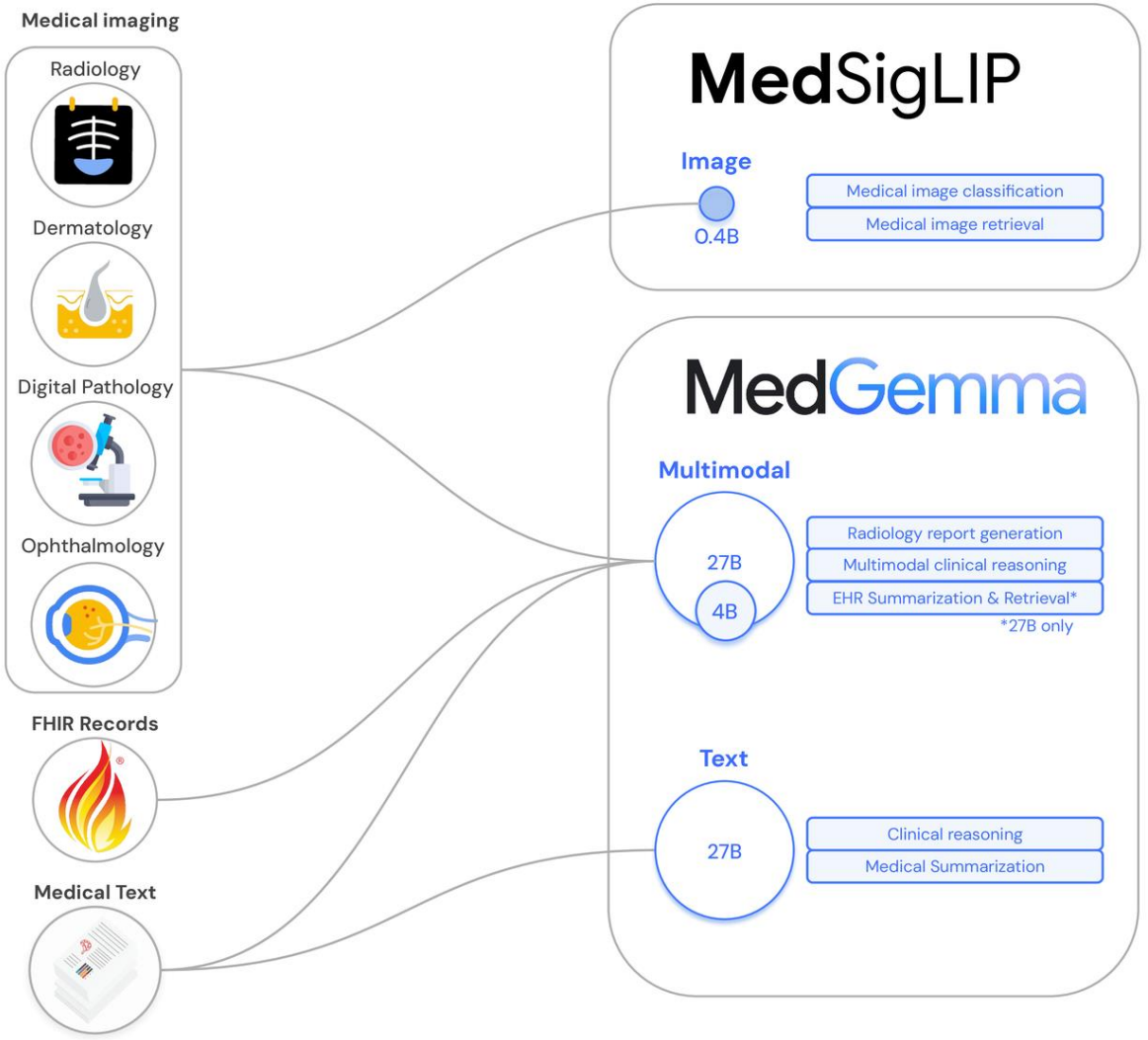
Why useful for us?

- Publicly accessible demos + code.
- Combines text + images in realistic healthcare tasks.

We will try out:

Radiology explainer, clinical decision support, and inference

(for fine-tuning a 40GB GPU is necessary)



UMC Utrecht

Medical models

MedGemma radiology demo

Task: Explain radiology images (e.g., chest X-rays, CTs) in natural language.

Features:

- Upload medical image → model generates explanation.
- Images and explanations have been pre-made; no interactive demo

Link:
https://huggingface.co/spaces/google/rad_explain

Medical models

MedGemma clinical decision support demo

Task: Provide recommendations from textual patient input.

Features:

- Combine clinical text (symptoms, history) with conversational data.
- Generate patient report.
- Develop diagnosis based on patient history and conversation.

Link:
https://huggingface.co/spaces/google/appoint-ready

Medical models

MedGemma – unstructured to structured data

Task: Convert unstructured free text data into a schematic representation

Features:

- Combine clinical text (symptoms, history) with conversational data.
- Generate patient report.
- Develop diagnosis based on patient history and conversation.

Link:
https://huggingface.co/spaces/google/radextract

Practical session 2

Using a VLM with code

MedGemma 4B or 27B on Google Colab

Features:

- Combine text and vision data
- Ask it any question you like!

Assignment:

Try out the notebook for yourself! Try at least the following:
- Loading and running the model with the provided data
- Try running it with a different prompt and image input from the internet

LINK TO NOTEBOOK

https://colab.research.google.com/github/google-health/medgemma/blob/main/notebooks/quick_start_with_hugging_face.ipynb

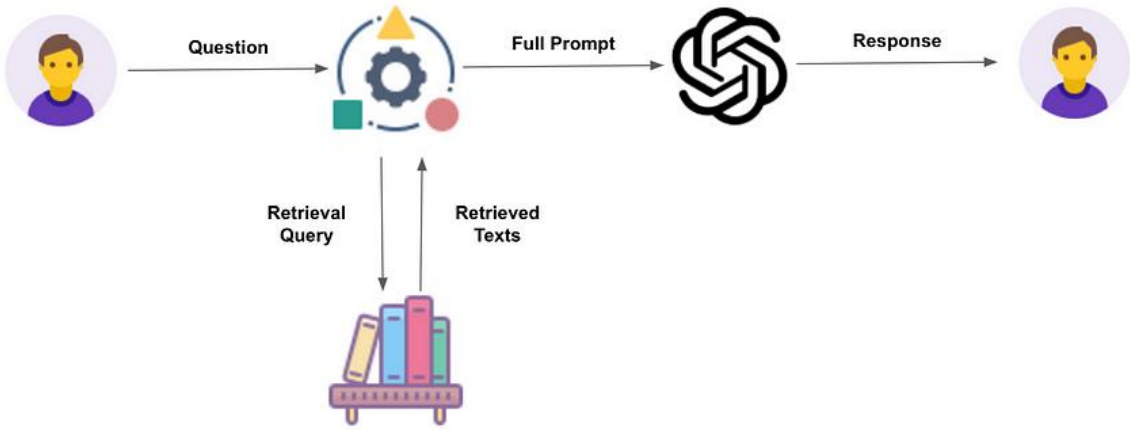
Lunch!

Retrieval Augmented Generation (RAG)

What is it?

Definition:
LLM augmented with access to an external knowledge base.

Query → Retrieve relevant documents → Generate response based on retrieved data.



Example:
A medical LLM that retrieves the latest updated medical knowledge from pubmed.

UMC Utrecht

Retrieval Augmented Generation (RAG)

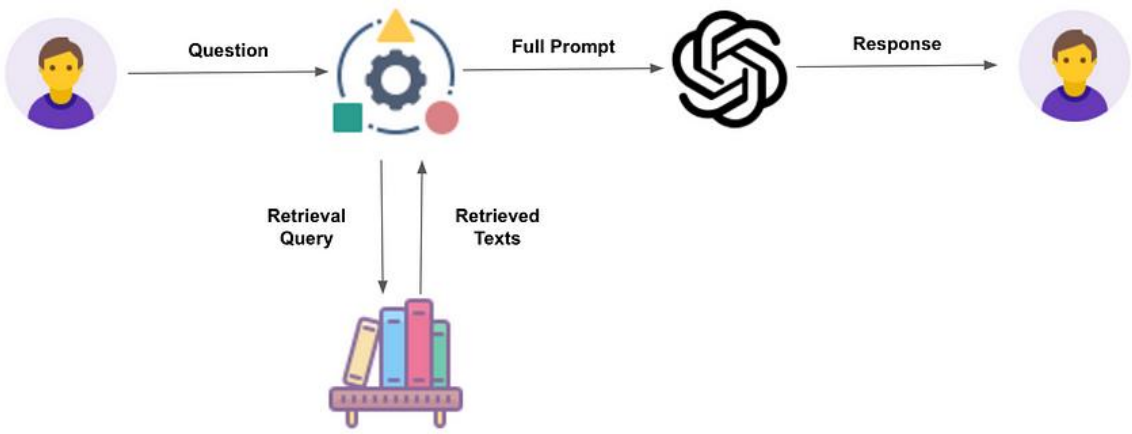
What is it?

Do it yourself:

- Construct database of data or ‘knowledge’.
- Set up retrieval system (text search or embedding vector search).
- Integrate LLM with retrieval pipeline (LangChain, Huggingface).

Existing RAG implementations:

- ChatGPT (or Claude, Bing, Gemini) searching the internet.
- ChatGPT plugins (Scholar.ai, Consensus, Scholar GPT).



Practical session 3

Using proprietary models with code

So far, we have only been using open-source models

These definitely have benefits, but often underperform the large models available online

Now you get a chance to play with proprietary models, for free!

LINK TO NOTEBOOK

Practical assignment

Validation experiment

Choose two models

- Can both be proprietary (e.g. ChatGPT (various versions), Anthropic, Claude, etc.)
- Can both be open-source (e.g. MedGemma, Llama, Qwen, etc.)
- Can be a proprietary and an open-source model

Choose a dataset

- Look on Kaggle, Huggingface, etc.
- Can be text, images, ECG/EEG, combinations of each

Choose a metric

- What do you want to measure?

Investigate: What can your model do? Where is it lacking? Which model performs best? What are the biases or risks inherent to the model (check documentation)