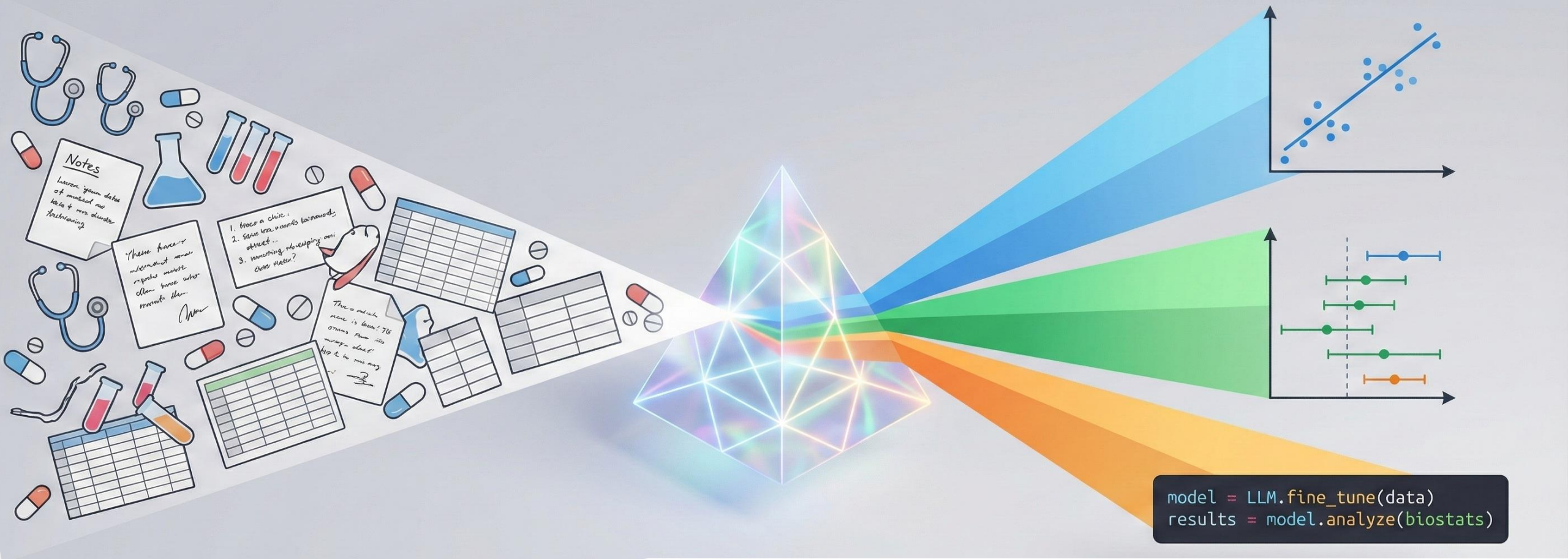


Large Language Models for Research

Datascience & Biostatistics

UMC Utrecht

Ruurd Kuiper – Julius Center, UMCU

AI will replace us!

Forbes

EDITORS' PICK | LEADERSHIP > CAREERS

These Jobs Will Fall First As AI Takes Over The Workplace

AI will replace us!

Forbes

EDITORS' PICK | LEADERSHIP > CAREERS

These Jobs Will Fall First As AI Takes Over The Workplace

Nearly 4 in 10 companies will replace workers with AI by 2026, survey shows

High-salary employees, those without AI skills, recently hired workers and entry-level employees face the highest risks for layoffs.

Published Sept. 22, 2025

AI will replace us!

Forbes

EDITORS' PICK | LEADERSHIP > CAREERS

These Jobs Will Fall First As AI Takes Over The Workplace

Nearly 4 in 10 companies will replace workers with AI by 2026, survey shows

High-salary employees, those without AI skills, recently hired workers and entry-level employees face the highest risks for layoffs.

Published Sept. 22, 2025

AI will replace us!

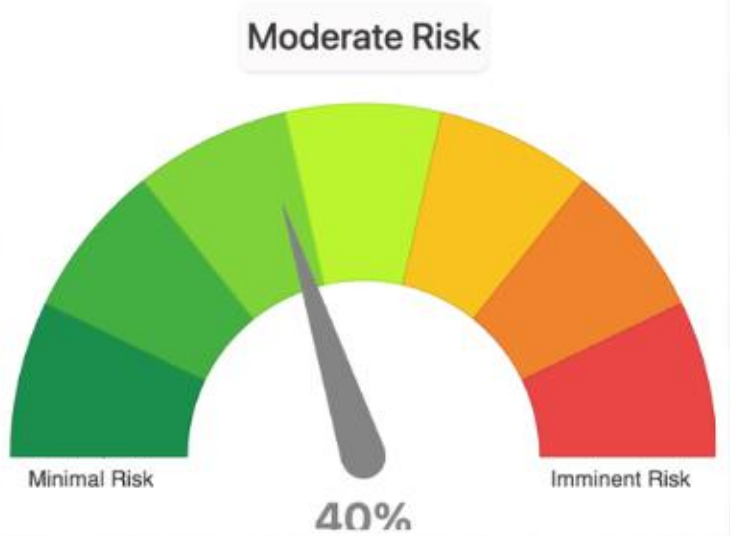
Forbes

EDITORS' PICK | LEADERSHIP > CAREERS

These Jobs Will Fall First As AI Takes Over The Workplace

WILL ROBOTS TAKE MY JOB?

Biostatisticians



The Death of the Data Scientist: What AI Tools Mean for Our Future

Manousos Kladis, PhD · Follow · 3 min read · Sep 16, 2025

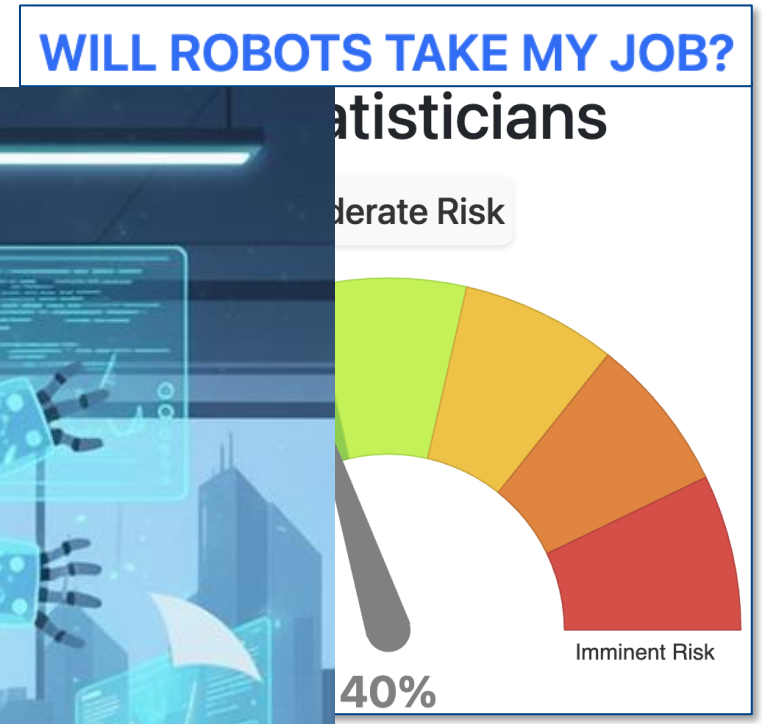
Nearly 4 in 10 companies will replace workers with AI by 2026, survey shows

High-salary employees, those without AI skills, recently hired workers and entry-level employees face the highest risks for layoffs.

Published Sept. 22, 2025

UMC Utrecht

AI will replace us!



UMC Utrecht

AI will replace us!

“If you work as a radiologist, you're like Wile E. Coyote in the cartoons that is already **over the edge of the cliff** but hasn't yet looked down. People should stop training radiologists now. It is completely obvious that **within five years** deep learning is going to do better than radiologists.”

Geoffrey Hinton (2016)

https://www.youtube.com/watch?v=2HMPRXstSvQ

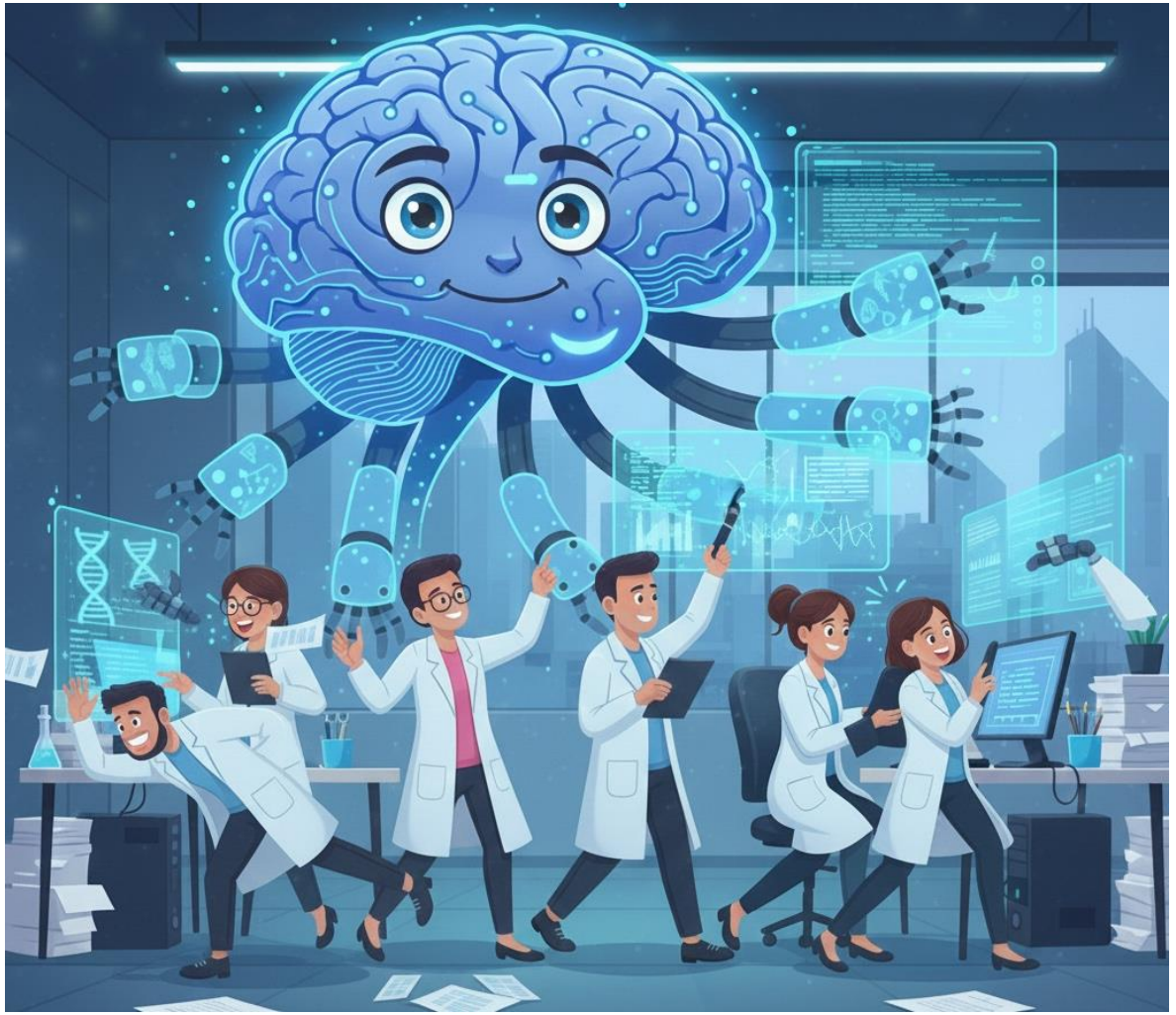
AI can assist us!

“Hinton said that he spoke too broadly in 2016. Furthermore, he said he was **wrong on the timing but not the direction**. In his new prediction, Hinton said that most medical image interpretation will be performed by a “**combination of A.I. and a radiologist**, and it will make radiologists a whole lot more efficient in addition to improving accuracy.”

Geoffrey Hinton (2025)
New York Times

AI can assist us!

AI can assist us!



AI can assist us!

If we let it...

BUSINESS INSIDER

AI

AI won't replace human workers, but 'people that use it will replace people that don't,' AI expert Andrew Ng says

By Lakshmi Varanasi + Follow

Andrew Ng is an AI optimist. Steve Jennings / Stringer/Getty Images

AI can assist us!

If we let it…

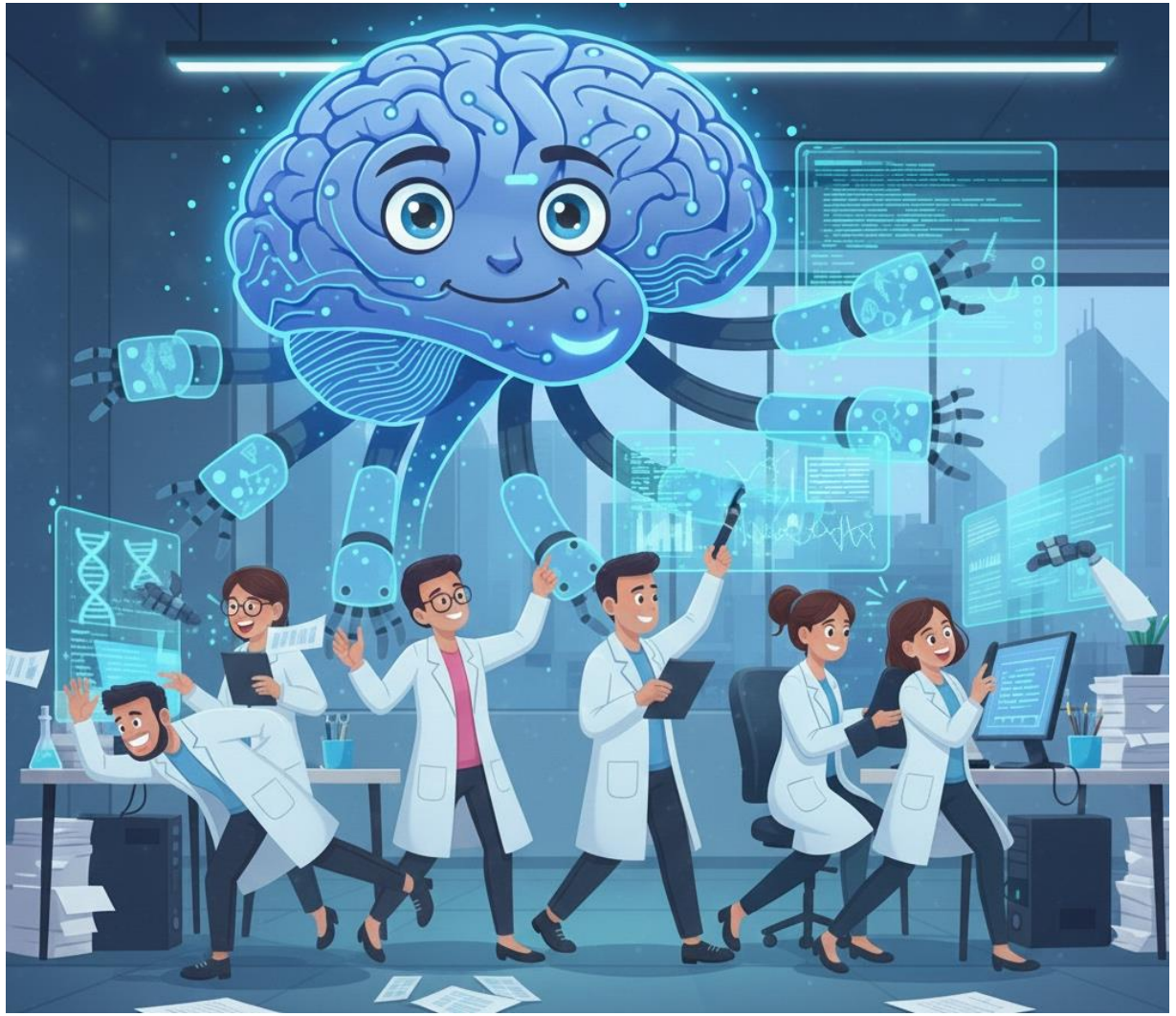
BUSINESS INSIDER

AI

AI won't replace human workers, but 'people that use it will replace people that don't,' AI expert Andrew Ng says

By Lakshmi Varanasi + Follow

AI can assist us!



AI can assist us!

- Literature **synthesis**
- Grant **writing**
- Protocol **drafting**
- Code **generation**
- Data **cleaning**
- Statistical **reporting**
- …

AI can assist us!

- Literature **synthesis**
- Grant w**riting**
- Protocol **drafting**
- Code **generation**
- Data **cleaning**
- Statistical **reporting**
- …

Are we there yet?

AI can assist us!

- Literature **synthesis**
- Grant w**riting**
- Protocol **drafting**
- Code **generation**
- Data **cleaning**
- Statistical **reporting**
- …

Are we there yet? That's what we'll find out today!

In which ways we are
In which ways we are not

AI can assist us!

- Literature **synthesis**
- Grant w**riting**
- Protocol **drafting**
- Code **generation**
- Data **cleaning**
- Statistical **reporting**
- …

Are we there yet? That's what we'll find out today!

In which ways we are
In which ways we are not

(How) can we use LLMs to our advantage?

Contents

Contents

Introduction to LLMs in Research

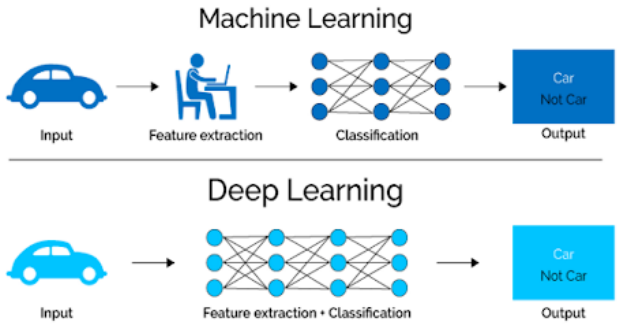
What are LLMs?

A **large language model** (LLM) is a **deep learning neural network** trained on **large (text) datasets** to learn **statistical patterns of language** enabling it to **generate text** by **predicting** the **mostly likely** next **token** in a sequence.

Introduction to LLMs in Research

What are LLMs?

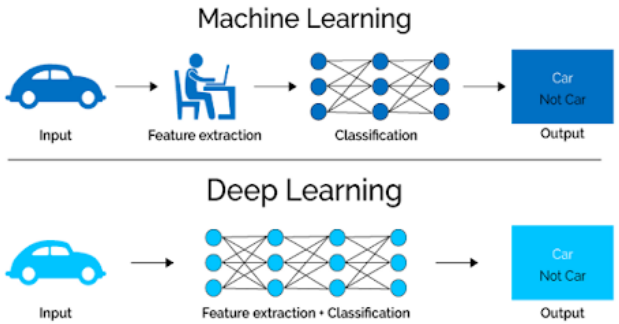
A **large language model** (LLM) is a **deep learning neural network** trained on **large (text) datasets** to learn **statistical patterns of language** enabling it to **generate text** by **predicting** the **mostly likely** next **token** in a sequence.

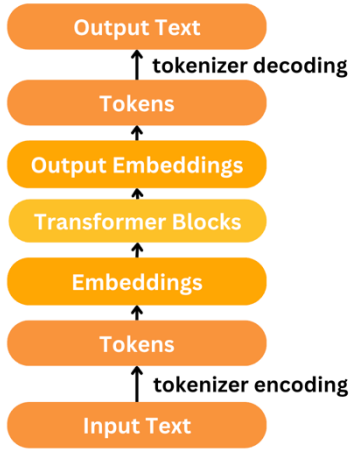


Introduction to LLMs in Research

What are LLMs?

A **large language model** (LLM) is a **deep learning neural network** trained on **large (text) datasets** to learn **statistical patterns of language** enabling it to **generate text** by **predicting** the **mostly likely** next **token** in a sequence.

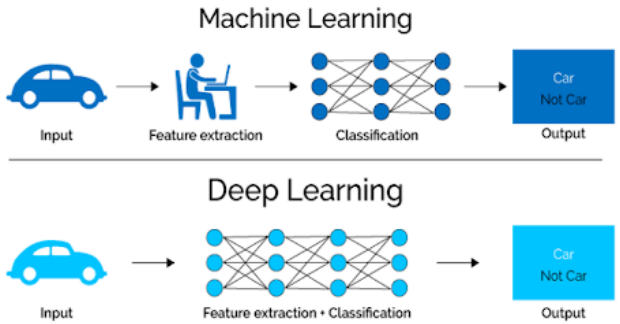


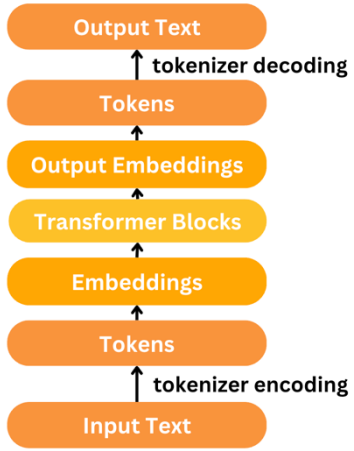


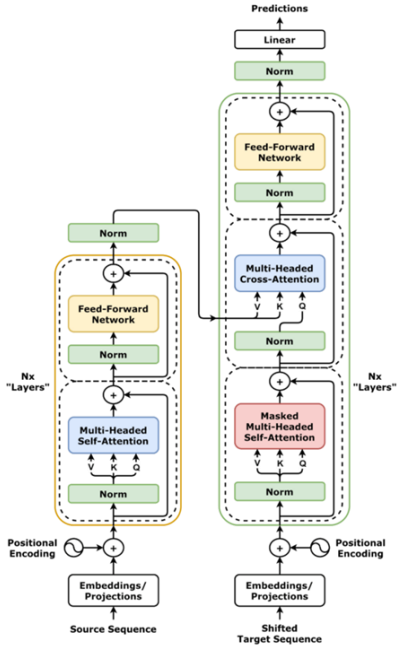
Introduction to LLMs in Research

What are LLMs?

A **large language model** (LLM) is a **deep learning neural network** trained on **large (text) datasets** to learn **statistical patterns of language** enabling it to **generate text** by **predicting** the **mostly likely** next **token** in a sequence.



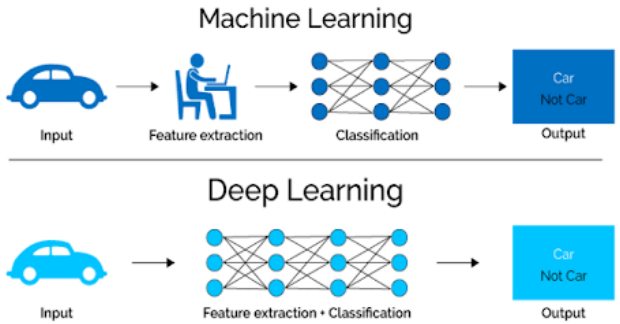


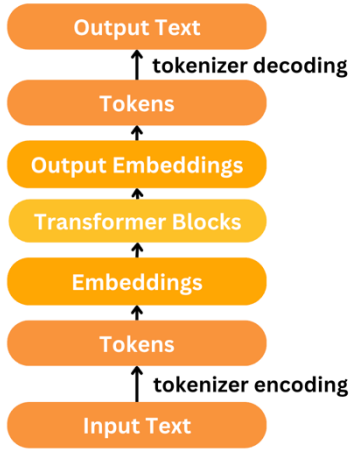


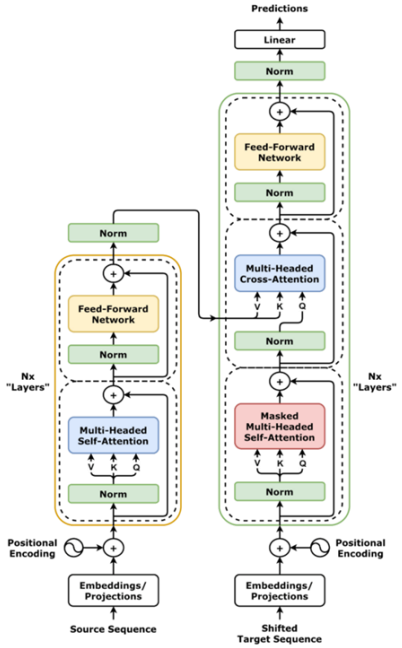
Introduction to LLMs in Research

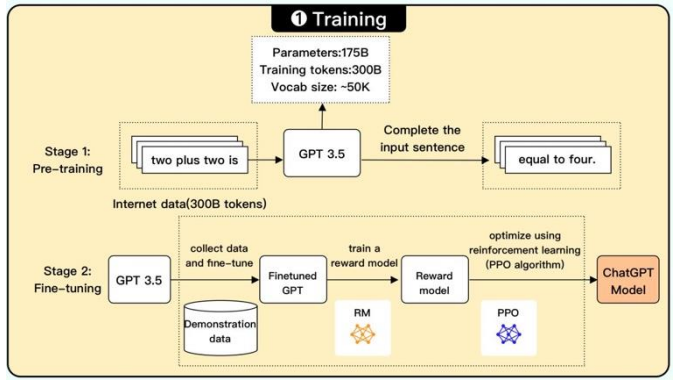
What are LLMs?

A **large language model** (LLM) is a **deep learning neural network** trained on **large (text) datasets** to learn **statistical patterns of language** enabling it to **generate text** by **predicting** the **mostly likely** next **token** in a sequence.









Introduction to LLMs in Research

What are LLMs?

A **large language model** (LLM) is a **deep learning neural network** trained on **large (text) datasets** to learn **statistical patterns of language** enabling it to **generate text** by **predicting** the **mostly likely** next **token** in a sequence.

LLMs generate text.

Introduction to LLMs in Research

What are LLMs?

A **large language model** (LLM) is a **deep learning neural network** trained on **large (text) datasets** to learn **statistical patterns of language** enabling it to **generate text** by **predicting** the **mostly likely** next **token** in a sequence.

LLMs generate text.

This enables them to be used for many applications.

Introduction to LLMs in Research

What are LLMs?

A **large language model** (LLM) is a **deep learning neural network** trained on **large (text) datasets** to learn **statistical patterns of language** enabling it to **generate text** by **predicting** the **mostly likely** next **token** in a sequence.

LLMs generate text.

This enables them to be used for many applications.

We will discuss some of those today!

Introduction to LLMs in Research

Capabilities – text generation

General task categories

- **Text summarization**
- **Question answering**
- **Code / data generation**
- **Translation / multilingual tasks**
- **Sentiment / opinion analysis**
- **Text classification / tagging**
- **Information retrieval / search**
- **Text completion / generation**
- **Dialogue / conversational agents** – Chatbots, assistants.
- **Reasoning / logic tasks** – Math problems, commonsense reasoning, planning.
- **Recommendation / ranking** – Suggesting items or ranking options.
- **Data-to-text / text-to-data** – Converting structured data into text and vice versa.
- …

Introduction to LLMs in Research

Capabilities – healthcare specific

Healthcare specific tasks

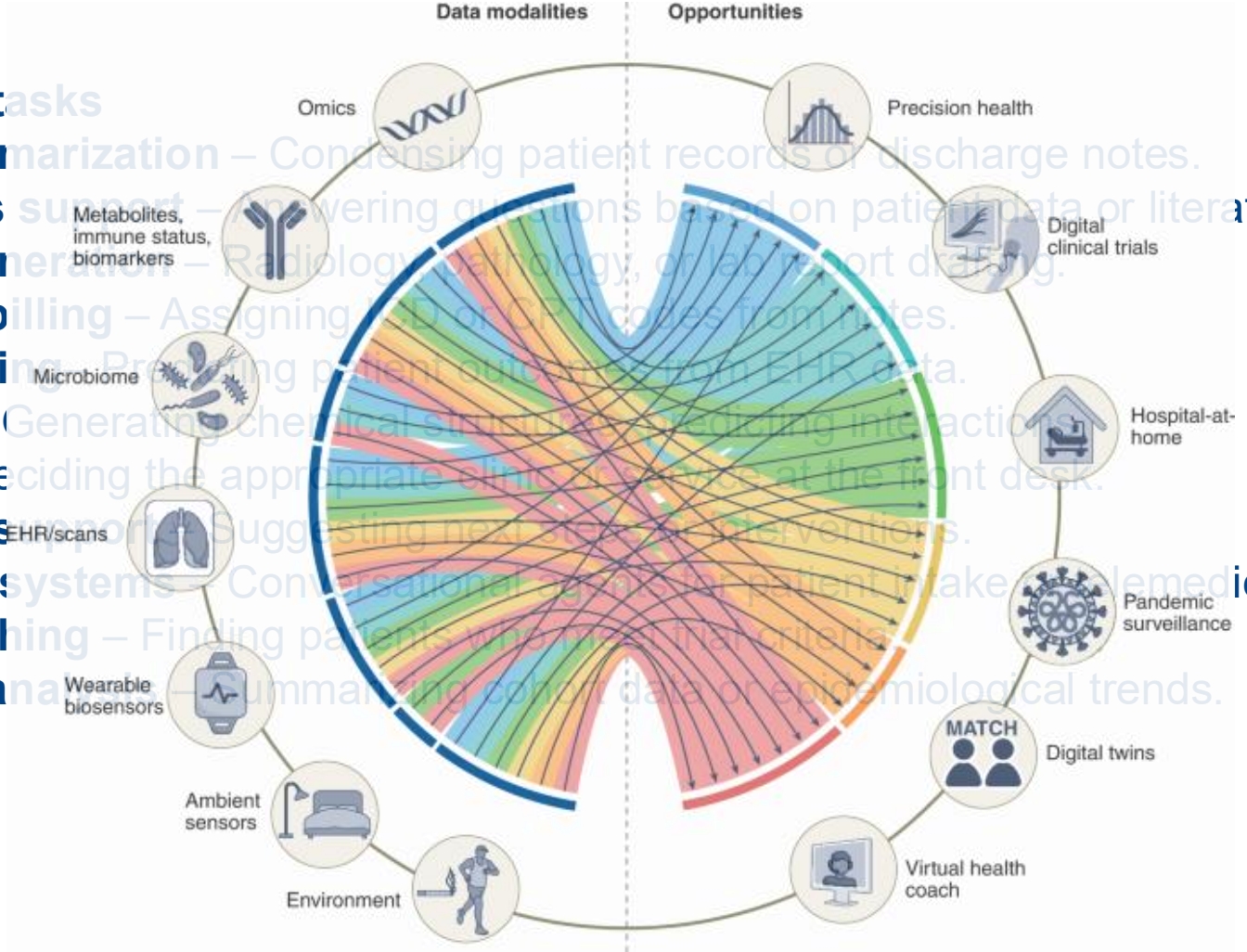
- **Clinical note summarization** – Condensing patient records or discharge notes.
- **Clinical diagnosis support** – Answering questions based on patient data or literature.
- **Medical report generation** – Radiology, pathology, or lab report drafting.
- **Clinical coding / billing** – Assigning ICD or CPT codes from notes.
- **Predictive modelling**– Predicting patient outcomes from EHR data.
- **Drug discovery** – Generating chemical structures, predicting interactions.
- **Patient triage** – Deciding the appropriate clinic or service at the front desk.
- **Clinical decision support** – Suggesting next steps or interventions.
- **Medical dialogue systems** – Conversational agents for patient intake or telemedicine.
- **Clinical trial matching** – Finding patients who meet trial criteria.
- **Population-level analysis** – Summarizing cohort data or epidemiological trends.

Introduction to LLMs in Research

Capabilities – healthcare specific

Healthcare specific tasks

- **Clinical note sum**marization – Condensing patient records or discharge notes.
- **Clinical diagnosis** support – Answering questions based on patient data or literature.
- **Medical report ge**neration – Radiology, pathology, or lab report drafting.
- **Clinical coding / b**illing – Assigning ICD or CPT codes from notes.
- **Predictive modelli**ng – Predicting patient outcomes from EHR data.
- **Drug discovery** – Generating chemical structures, predicting interactions.
- **Patient triage** – Deciding the appropriate clinic or service at the front desk.
- **Clinical decision s**upport – Suggesting next steps or interventions.
- **Medical dialogue s**ystems – Conversational agents for patient intake, telemedicine.
- **Clinical trial match**ing – Finding patients who meet trial criteria.
- **Population-level a**nalysis – Summarizing cohort data or epidemiological trends.

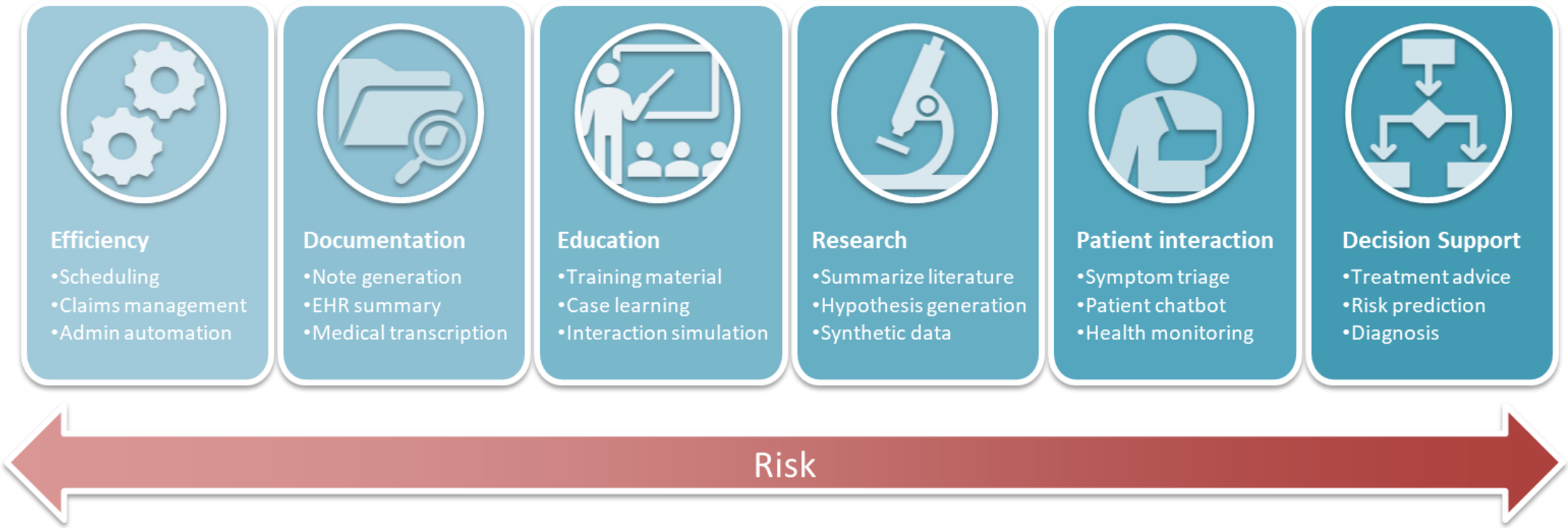


Acosta, J.N., Falcone, G.J., Rajpurkar, P. *et al.* Multimodal biomedical AI. *Nat Med* **28**, 1773–1784 (2022). https://doi.org/10.1038/s41591-022-01981-2

Introduction to LLMs in Research

Capabilities – healthcare specific

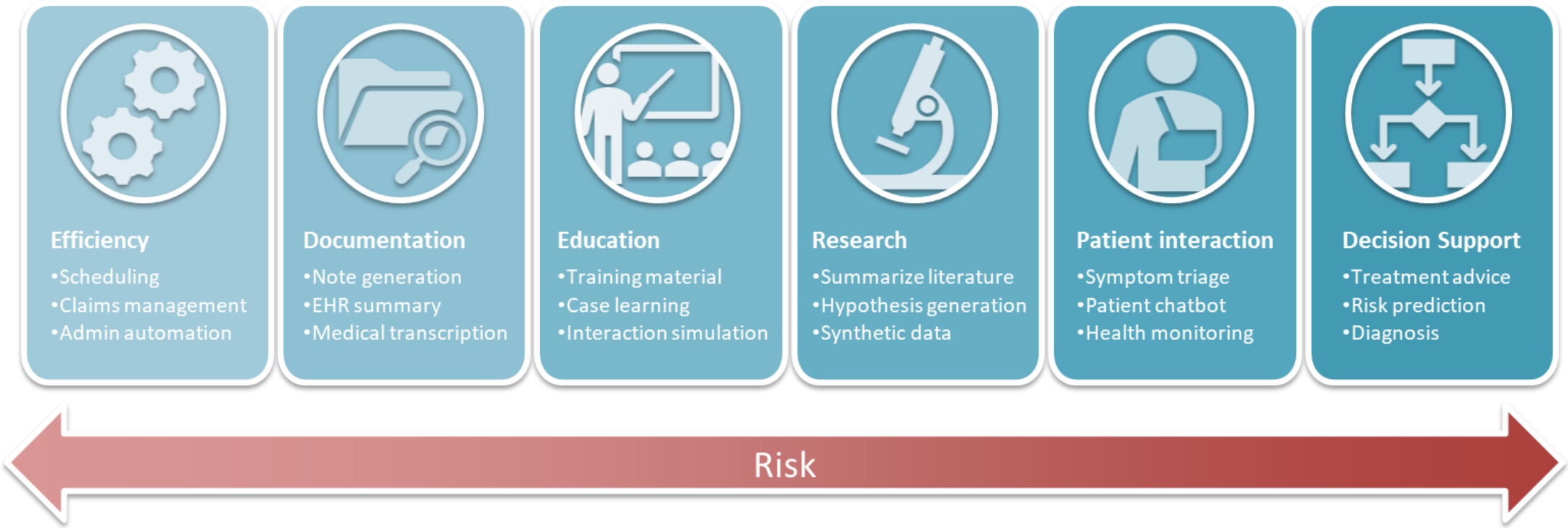
However, each application carries a certain risk



Introduction to LLMs in Research

Capabilities – healthcare specific

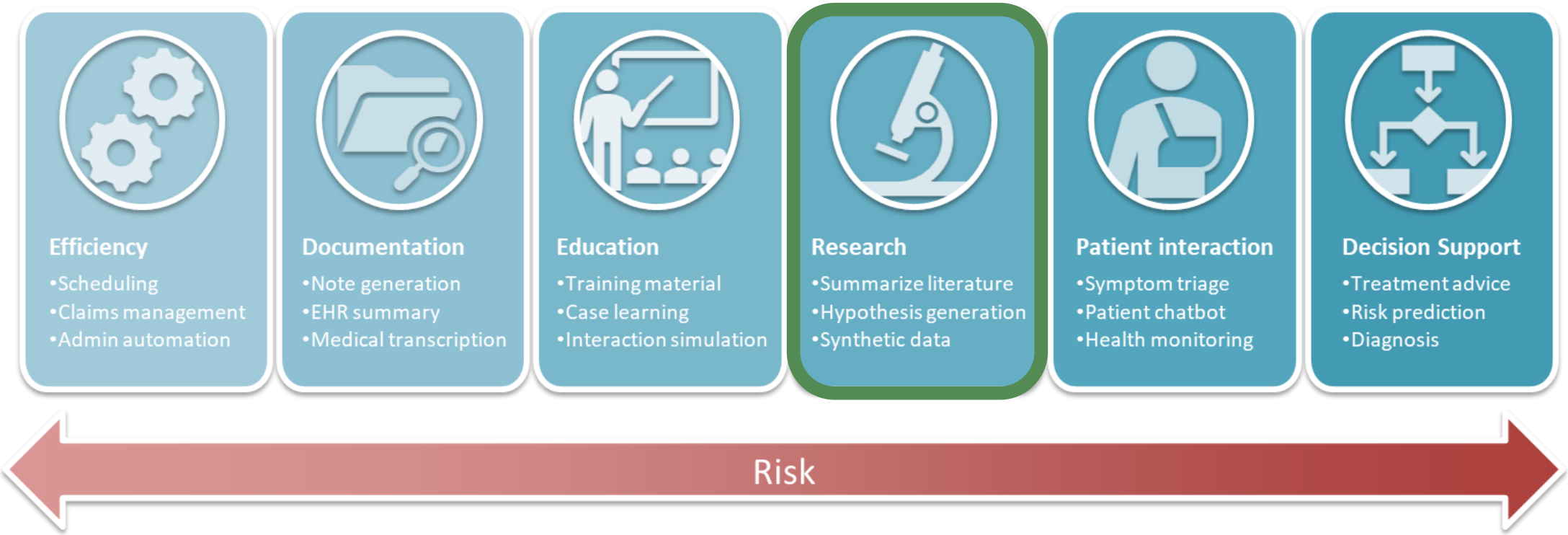
Especially in healthcare, we should take these risks into account…



Introduction to LLMs in Research

Capabilities – healthcare specific

Especially in healthcare, we should take these risks into account…



Use in research is quite risky!

Introduction to LLMs in Research

Capabilities – research specific

Some research specific tasks (that we will discuss):

- Literature triage → information extraction → synthesis → drafting
- Code generation → troubleshooting → translation → testing
- Data pre-processing → extraction → harmonisation → reporting

But before we do so…

Current capabilities and limitations

Challenges

General challenges:

Biases: From data but also from the user
Sycophancy: LLMs like to agree with you
Privacy: Training data, or use of proprietary (cloud) models
Cost: Savings may not justify costs
Ethical: Bias, accuracy, job-replacement, accountability, transparency

LLM-specific challenges:

Hallucinations: Content that is false (or unfaithful to the provided source content)
Omissions: Content that should be known or is present in the context but is omitted
Trivial information: Information that is true, but not necessary

Some guidelines

Some tips (reiterated)

When not to use LLMs:

- **Fact retrieval:** they generate plausible text, not guaranteed truth.
- **Sensitive or confidential data:** risk of data leakage, especially with cloud models.
- **Clinical decision-making:** unsafe without expert oversight.

When LLMs can be useful:

- **Rewriting and reformatting:** improving clarity, tone, style or removing (grammatical) errors
- **Summarizing:** For research but also in the clinic
- **Brainstorming or ideation**
- **Structuring information:** turning free text into tables, or one tabular format into another.
- **Drafting non-critical text:** e.g., educational materials, reports, or patient communication.

Moral of the story:

- ***Treat LLMs as language assistants, not knowledge authorities.***
- ***Keep thinking for yourself!***
  Two brains are better than one, but an e-brain is (still) worse than a real one!

Applications of LLMs in Research

Note!

What LLMs are capable of changes day by day

Applications of LLMs in Research

Note!

What LLMs are capable of changes day by day

Be sure to **check** all LLM output

Applications of LLMs in Research

Note!

What LLMs are capable of changes day by day

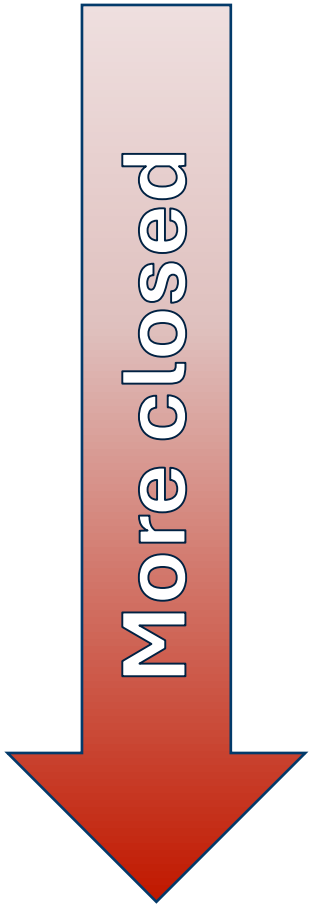
Be sure to **check** all LLM output

Judge for yourself what you think is appropriate
(unless there are rules or guidelines for your use case, then follow those)

Contents

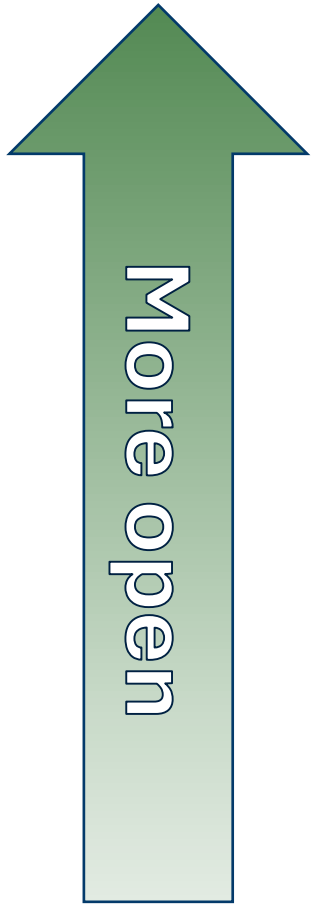
Proprietary vs open-source

Different levels



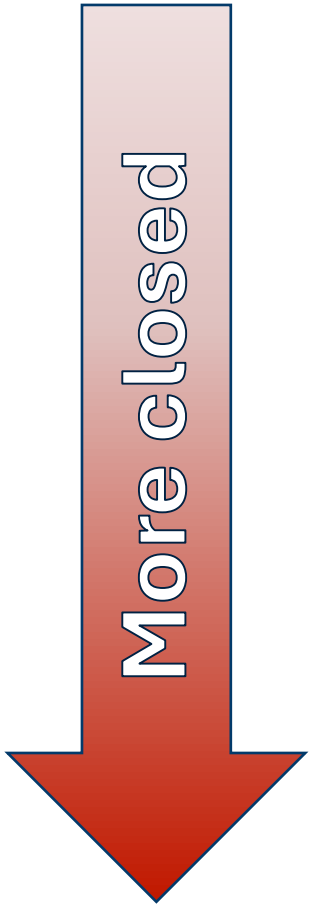
1. Fully Open (Weights, Code, and Data) (Pythia, RedPajama)

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



Proprietary vs open-source

Different levels

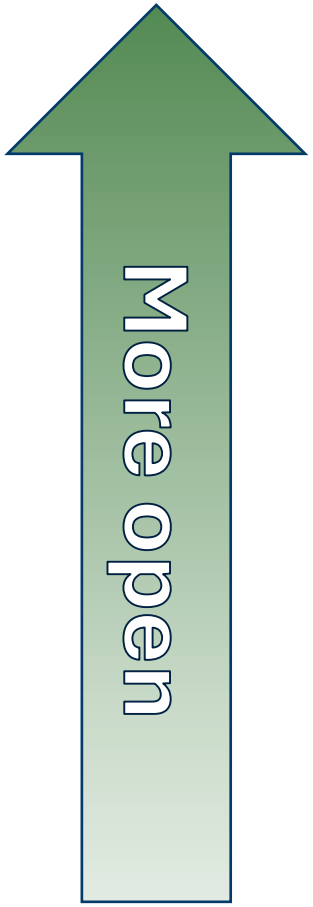


1. Fully Open (Weights, Code, and Data) (Pythia, RedPajama)

- **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



Proprietary vs open-source

Different levels

More closed

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

More open

Proprietary vs open-source

Different levels

More closed

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

More open

Proprietary vs open-source

What are the trade-offs?

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

Proprietary vs open-source

What are the trade-offs?

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.

Proprietary vs open-source

What are the trade-offs?

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

Proprietary vs open-source

What are the trade-offs?

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.

What model to use?

Size matters

Small Models (1-10B parameters)

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

What model to use?

Size matters

Small Models (1-10B parameters)

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

Medium Models (10-70B parameters)

- Workstation GPUs (24-100GB VRAM)
- Local GPU with quantization
- Multi-GPU for full precision
- Cloud-based deployment
- Cost efficiency: $0.20-1.00/hour

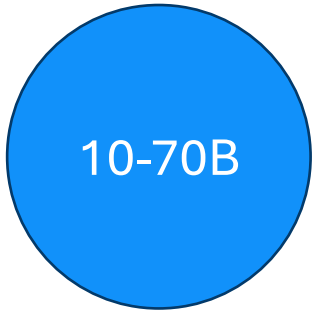
What model to use?

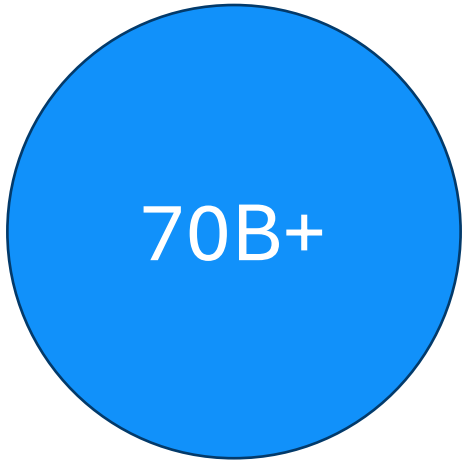
Size matters

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

Medium Models (10-70B parameters)
- Workstation GPUs (24-100GB VRAM)
- Local GPU with quantization
- Multi-GPU for full precision
- Cloud-based deployment
- Cost efficiency: $0.20-1.00/hour

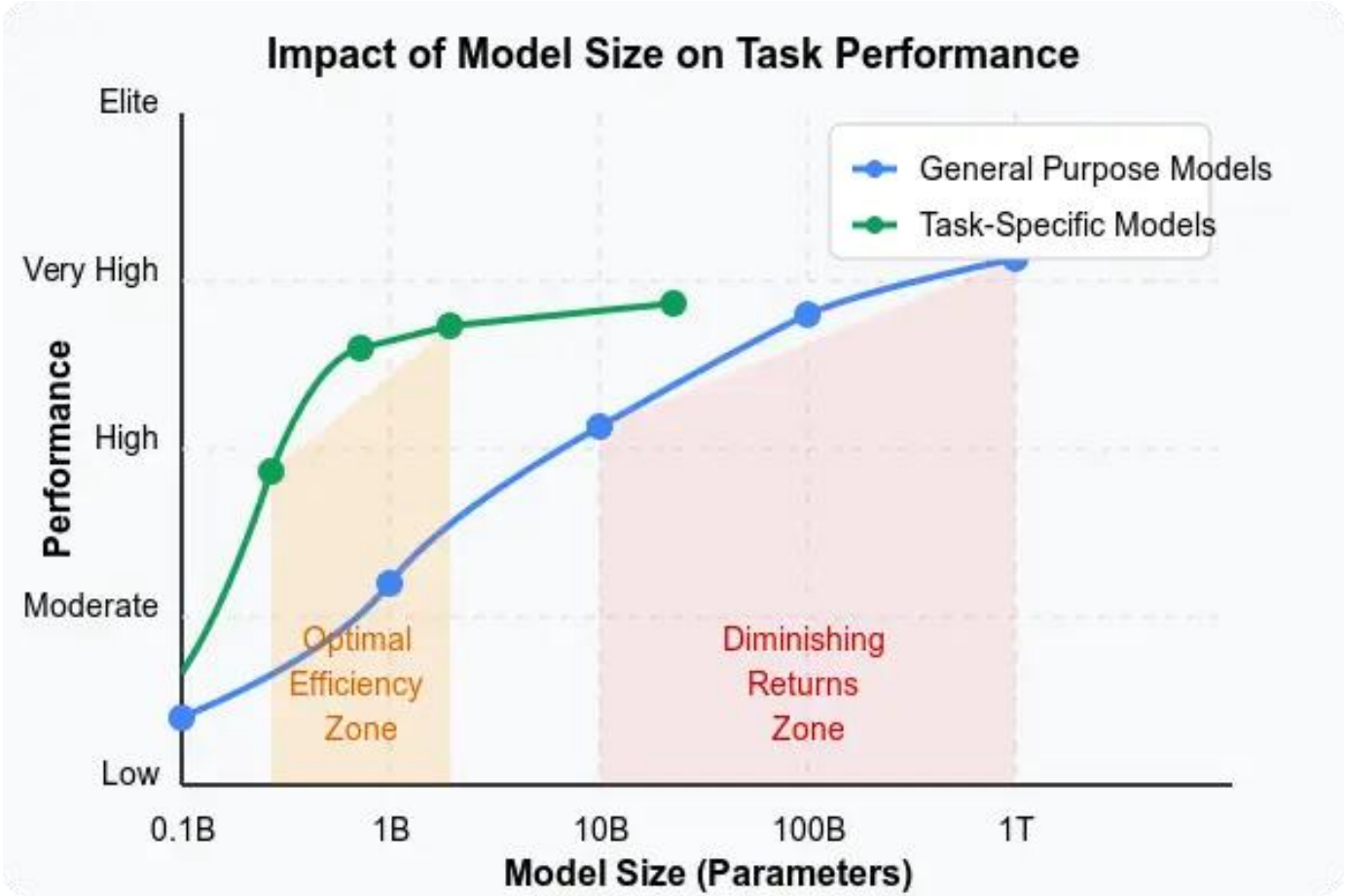
Large Models (70B+ parameters)
- Data center GPUs (80GB+ VRAM)
- Multiple high-end GPUs (A100, H100)
- Distributed computing
- Specialized AI cloud services
- Cost efficiency: $1.50-10.00+/hour





What model to use?

Size matters… But it's also about how you use it!



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

General vs (medical) task specific

Pros and cons

General LLMs

Pros:

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

Dedicated models

Pros:

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

General vs (medical) task specific

Pros and cons

General LLMs

Pros:
- Huge training data -> broad knowledge
- Strong 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.
- Better alignment with clinical tasks.

Cons:
- Narrower knowledge base.
- Smaller training sets -> risk of bias, overfitting.
- Less commercial support than general LLMs.
- Weaker general reasoning

Contents

But first: what do you think?

Mentimeter

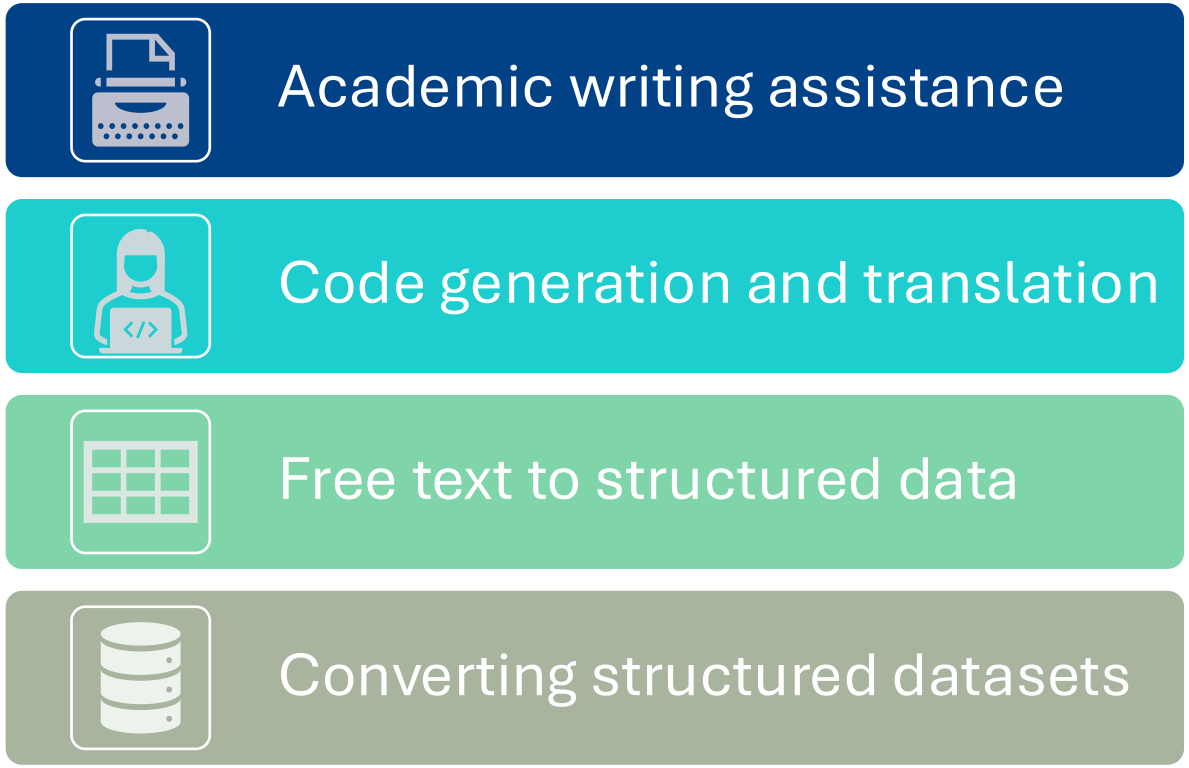
https://www.menti.com/al8w7c6ytczw

8910 4883

Applications of LLMs in Research

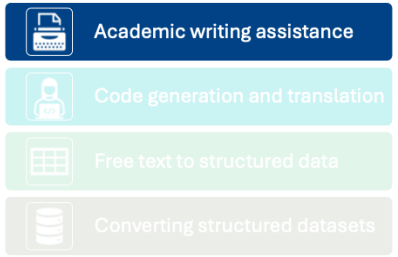
Different applications

We'll be discussing four topics:



LLMs for writing assistance

The task



Why use LLMs?

- Academic writing takes up a lot of time
- Structure can often be quite similar
- A lot of writing is already quite automatic
- Different people have different levels of language skills

LLMs for writing assistance

The task

Academic writing assistance
Code generation and translation
Free text to structured data
Converting structured datasets

Why use LLMs?

- Academic writing takes up a lot of time
- Structure can often be quite similar
- A lot of writing is already quite automatic
- Different people have different levels of language skills

How LLMs can help

- **Ideation**
- Creating initial structure
- Maintaining consistent voice
- Helping to summarize complex literature
- Formatting for journals
- Adhering to grant or protocol guidelines
- As a sparring partner
- Levelling the language playing field
- Final checks

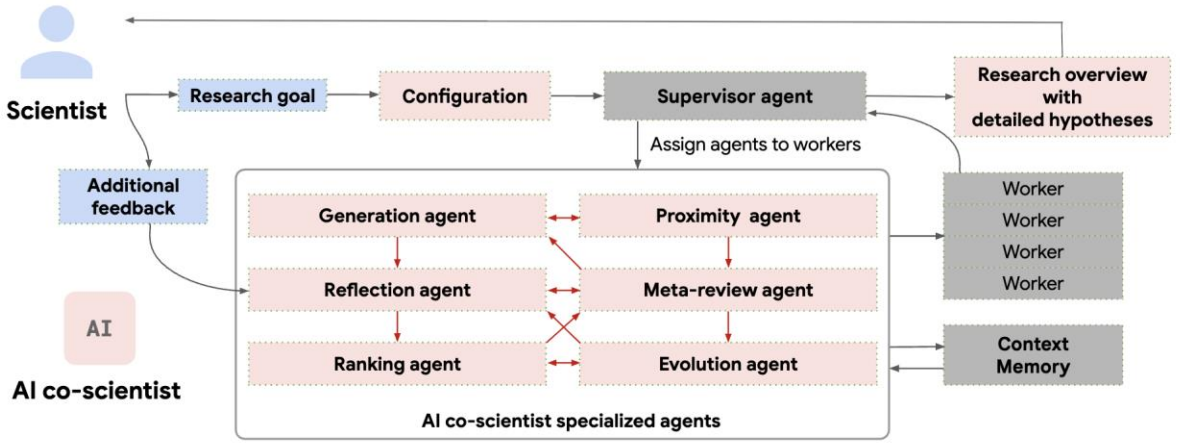
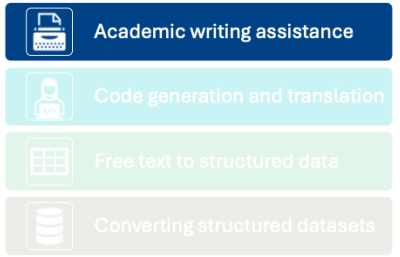


Figure 2 | The AI co-scientist multi-agent architecture design. The co-scientist accepts a natural language research goal from the user and parses this into a research plan configuration. This plan is then dispatched to the Supervisor agent which evaluates this plan to assigns weights and resources to each specialized agent and subsequently queues them as worker processes in a task queue according to these weights. The worker processes execute the queue of agent actions, and the system ultimately aggregates all information to formulate a research overview with detailed hypotheses and proposals for the scientist. The red boxes in the "The AI co-scientist specialized agents" section denote individual agents each with their own unique logic and role. The blue boxes indicate the scientist-in-the-loop inputs and feedback. The dark gray arrows represent the information flow through the co-scientist system, while the red arrows represent the information feedback loop between the specialized agents.

AI co-scientist

LLMs for writing assistance

The task

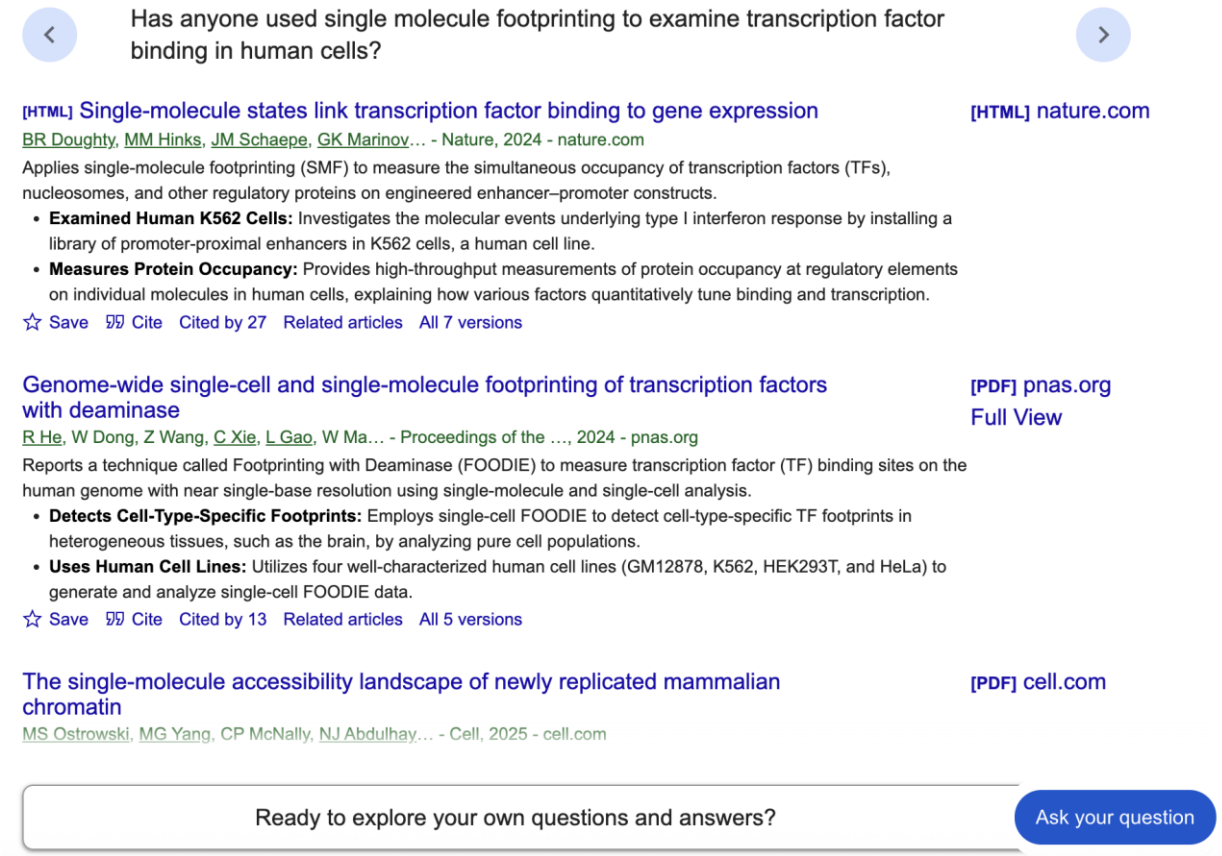


Why use LLMs?

- Academic writing takes up a lot of time
- Structure can often be quite similar
- A lot of writing is already quite automatic
- Different people have different levels of language skills

How LLMs can help

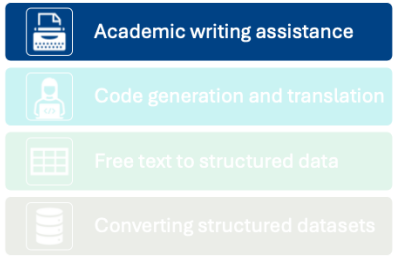
- Creating initial structure
- Maintaining consistent voice
- **Helping to summarize complex literature**
- Formatting for journals
- Adhering to grant or protocol guidelines
- As a sparring partner
- Levelling the language playing field
- Final checks



Scholar Labs

LLMs for writing assistance

How LLMs can and cannot be used



Future directions

- Automated ideation
- Fully automated (agentic) writing
- Automatic information synthesis

Safety checks

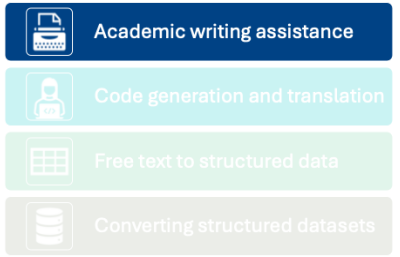
- Always verify claims manually
- Avoid invented citations entirely
- Especially check numbers
- Follow institutional data rules
- Follow authorship standards and document AI use
- Never use it for reviewing!

Work iteratively

- Use short prompts for control
- Ask to-the-point questions
- Check all output

LLMs for writing assistance

How LLMs can and cannot be used



Future directions

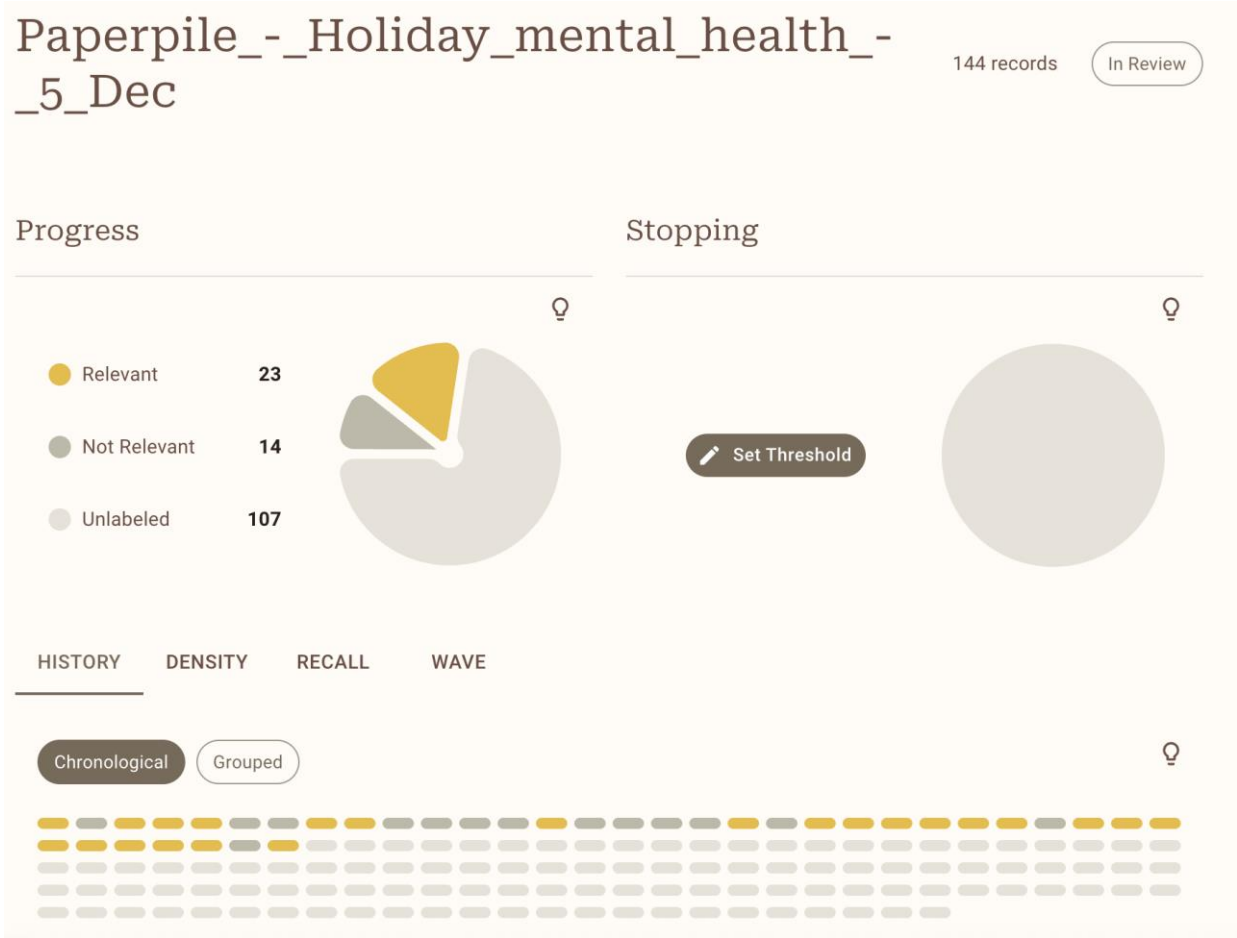
- Automated ideation
- Fully automated (agentic) writing
- Automatic information synthesis

Safety checks

- Always verify claims manually
- Avoid invented citations entirely
- Especially check numbers
- Follow institutional data rules
- Follow authorship standards and document AI use
- Never use it for reviewing!

Work iteratively

- Use short prompts for control
- Ask to-the-point questions
- Check all output

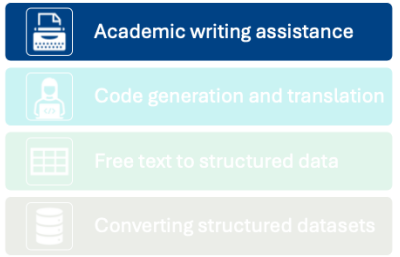


AS Review

UMC Utrecht

LLMs for writing assistance

How LLMs can and cannot be used



Future directions
- Automated ideation
- Fully automated (agentic) writing
- Automatic information synthesis

Safety checks
- Always verify claims manually
- Avoid invented citations entirely
- Especially check numbers
- Follow institutional data rules
- Follow authorship standards and document AI use
- Never use it for reviewing!

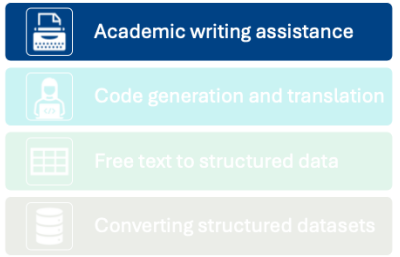
Work iteratively
- Use short prompts for control
- Ask to-the-point questions
- Check all output

A realistic workflow might be:
1. Retrieve papers from scholar/pubmed/…
2. Screen papers using AS Review
3. Extract information by conversing
4. Organize papers based on topic
5. Synthesize and draw preliminary conclusions
6. Draft rough outline of paper
7. Ideate about what you want to say in each paragraph
8. Write a rough draft of the paragraph including citations
9. Update writing style with LLM
10. Clean up your document
11. Check for grammatical, style and journal-specific requirements using LLM
12. CHECK EVERYTHING

LLM use possible (but check each answer!)

LLMs for writing assistance

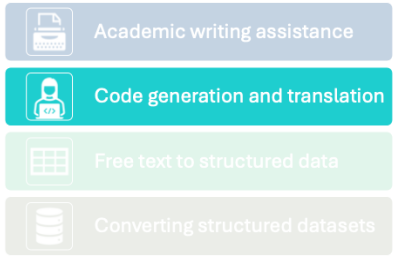
Some of the currently available tools



Tool	Best For...	Key Features	Limitations	Pricing
Google Scholar (Labs + PDF Reader)	**Reading & Navigation**	**PDF Reader (Extension):** Adds AI outlines, clickable citations, and figure jumping to any PDF in Chrome. **Labs:** New experimental "AI Search" that answers complex research questions with synthesized snippets.	Still experimental. The summaries are decent but strictly a reading aid, not a writer.	**Free**
SciSpace (typeset.io)	**Literature Review**	**Chat with PDF:** Explain math, tables, and text. **Lit Review Matrix:** Compare 10+ papers in a grid (methods, results, etc.). **Copilot:** Chrome extension to explain text on any website.	AI summaries can sometimes be generic. Free version is very limited.	**Freemium** (Premium ~$12/mo)
Consensus	**Fact-Checking & Evidence**	**Scientific Search:** Only searches peer-reviewed papers. **Yes/No Meter:** Visualizes scientific consensus on a question. **Synthesize:** Combines all information fully automatically	Not for creative writing or brainstorming. Strictly for a certain type of systematic reviews.	**Freemium** (Free: limited deep searches; Pro: ~$10-15/mo)
Perplexity	**Quick Research & Sourcing**	**Answer Engine:** Like Google + ChatGPT. Gives a direct answer with footnotes. **Pro Search:** Uses multi-step reasoning to find harder-to-locate details. **Focus Mode:** Restrict search to "Academic" sources only.	Citations in "standard" mode can include non-academic blogs/news. Less deep analysis than SciSpace.	**Freemium** (Pro: $20/mo)
ASReview	**Systematic Screening**	**Active Learning:** You screen ~1-10% of papers, AI learns your criteria and screens the rest. **Open Source:** Runs locally on your machine (data privacy). **Transparency:** fully reproducible steps.	Requires installation. It is a screener, not a summarizer/writer.	**Free** (Open Source)
Gemini (Deep Research)	**Deep Synthesis & Reporting**	**Deep Research:** Autonomous agent that browses hundreds of sites to write a long report (10+ pages). **Context Window:** Can analyze relatively large amounts of uploaded PDFs/Data (2M tokens).	"Deep Research" usually takes minutes to run. Can be "too much" for simple questions.	**Paid** (Part of Gemini Advanced: $20/mo)
ChatGPT (Deep Research)	**Autonomous Deep Dives**	**Deep Research:** Performs multi-step browsing, creates a research plan, and writes a cited report.	Availability of "Deep Research" varies by tier. "Pro" tier is expensive; "Plus" has lower caps.	**Paid** (Plus: $20/mo; Pro: $200/mo)
Jenni AI	**Drafting & Writing**	**Autocomplete:** Finishes your sentences based on your specific research context. **Citation Engine:** Suggests citations while you write. **Chat:** Chat with your library while drafting.	Can hallucinate text if you don't monitor it. Formatting exports can sometimes be tricky.	**Freemium** (Free is limited; Unlimited: ~$12-20/mo)

Code generation and conversion

The task

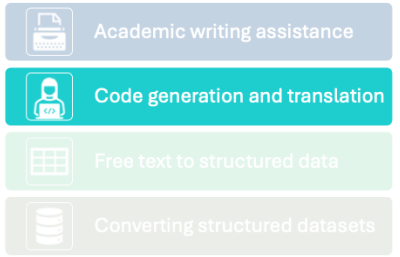


Why use LLMs?

- Coding requires a lot of human effort
- Unfamiliarity with syntax
- Debugging can be time-consuming
- Documentation, unit-testing etc. can be repetitive
- Style adherence

Code generation and conversion

The task

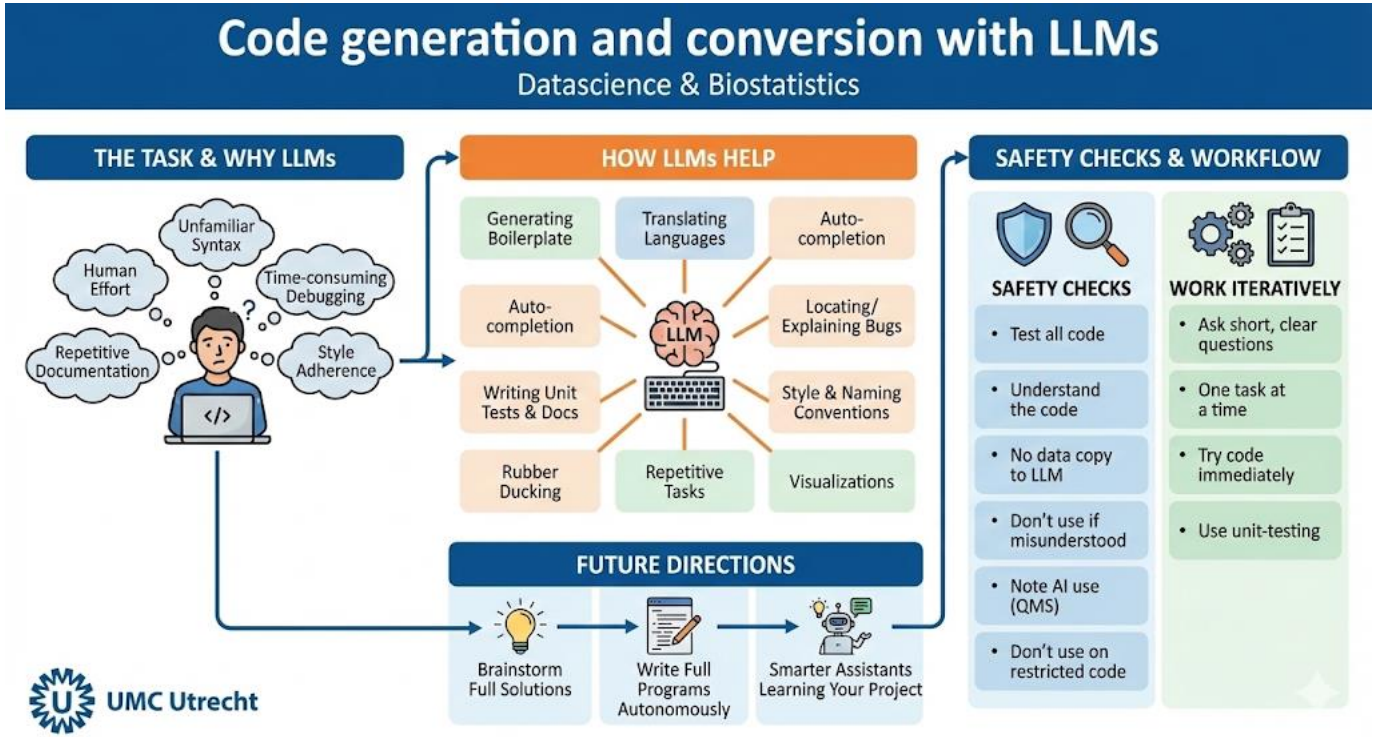


Why use LLMs?

- Coding requires a lot of human effort
- Unfamiliarity with syntax
- Debugging can be time-consuming
- Documentation, unit-testing etc. can be repetitive
- Style adherence

How LLMs can already help

- Generating boilerplate code
- Translating between languages
- Auto-completion for syntax
- Locating and explaining bugs
- Writing unit tests and documentation
- Helping with style and conventions
- Helping with naming variables
- Acting as a ‘rubber duck’
- Use for repetitive tasks
- Helping with visualizations



UMC Utrecht

Code generation and conversion

Existing tools

Academic writing assistance

Code generation and translation

Free text to structured data

Converting structured datasets

Future directions

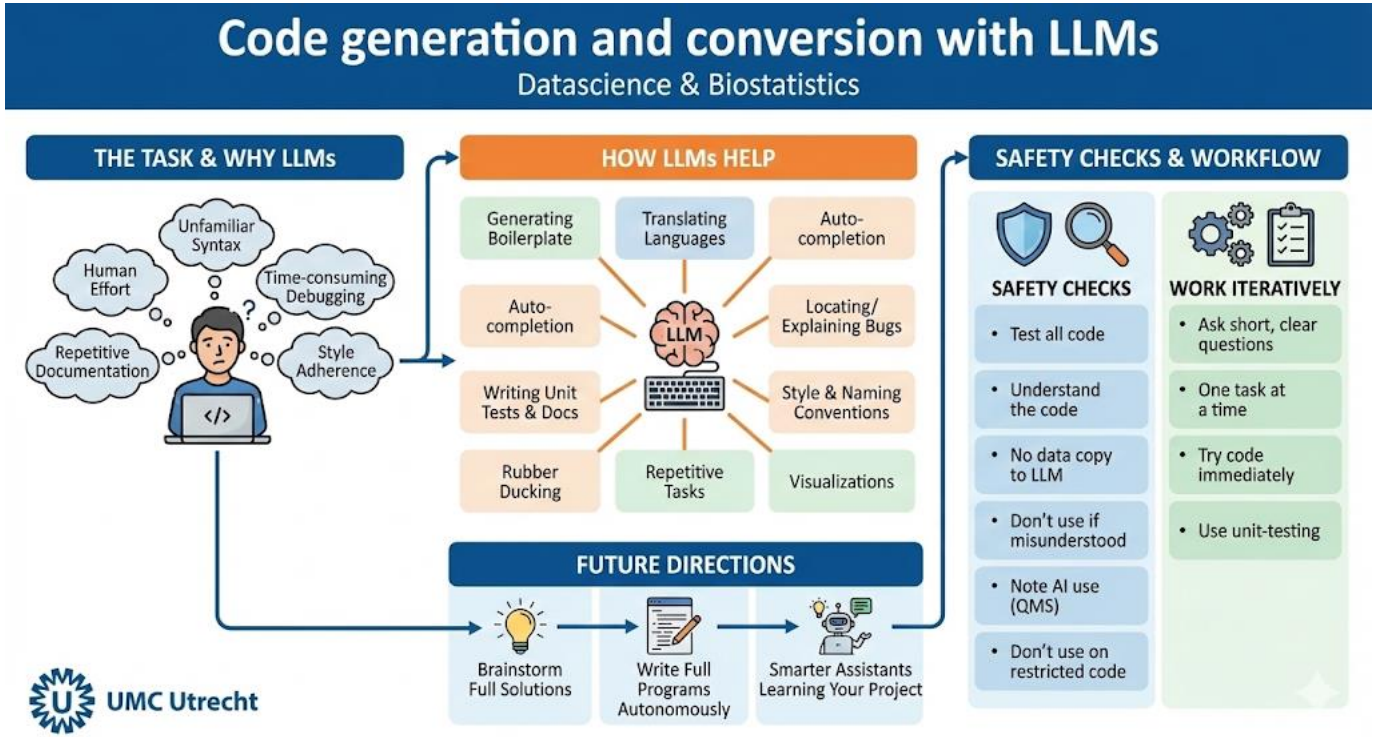
- Tools that help brainstorm full solutions
- Systems that write full programs on their own
- Smarter assistants that learn your project over time

Safety checks

- Test every piece of code
- Check if you understand the code
- Do not copy sensitive data to the LLM
- Do not use code you do not understand
- Note when and how AI was used (QMS)
- Do not use AI on sensitive code

Work iteratively!

- Ask short, clear questions
- Let the model handle one task at a time
- Try the code immediately
- Use unit-testing



UMC Utrecht

Code generation and conversion

Tools

Academic writing assistance

Code generation and translation

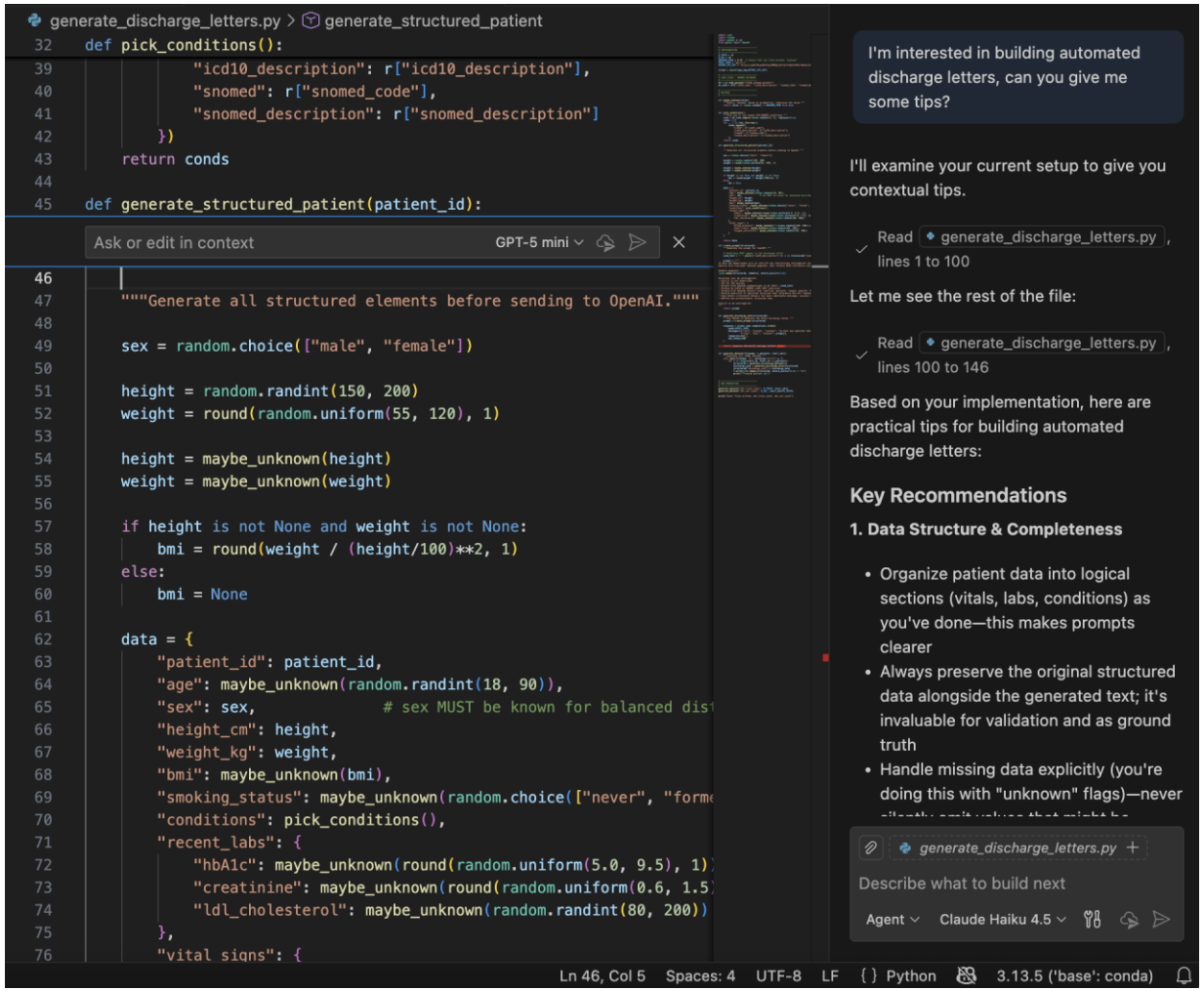
Free text to structured data

Converting structured datasets

Chatbased
- ChatGPT
- **Claude**
- Gemini
- Etc.

Integrated in IDE:
- Github Co-pilot
- GPT Codex
- Windsurf

AI-first IDE:
- Cursor
- …



VS code with Github Co-Pilot

UMC Utrecht

Code generation and conversion

Tools

Academic writing assistance

Code generation and translation

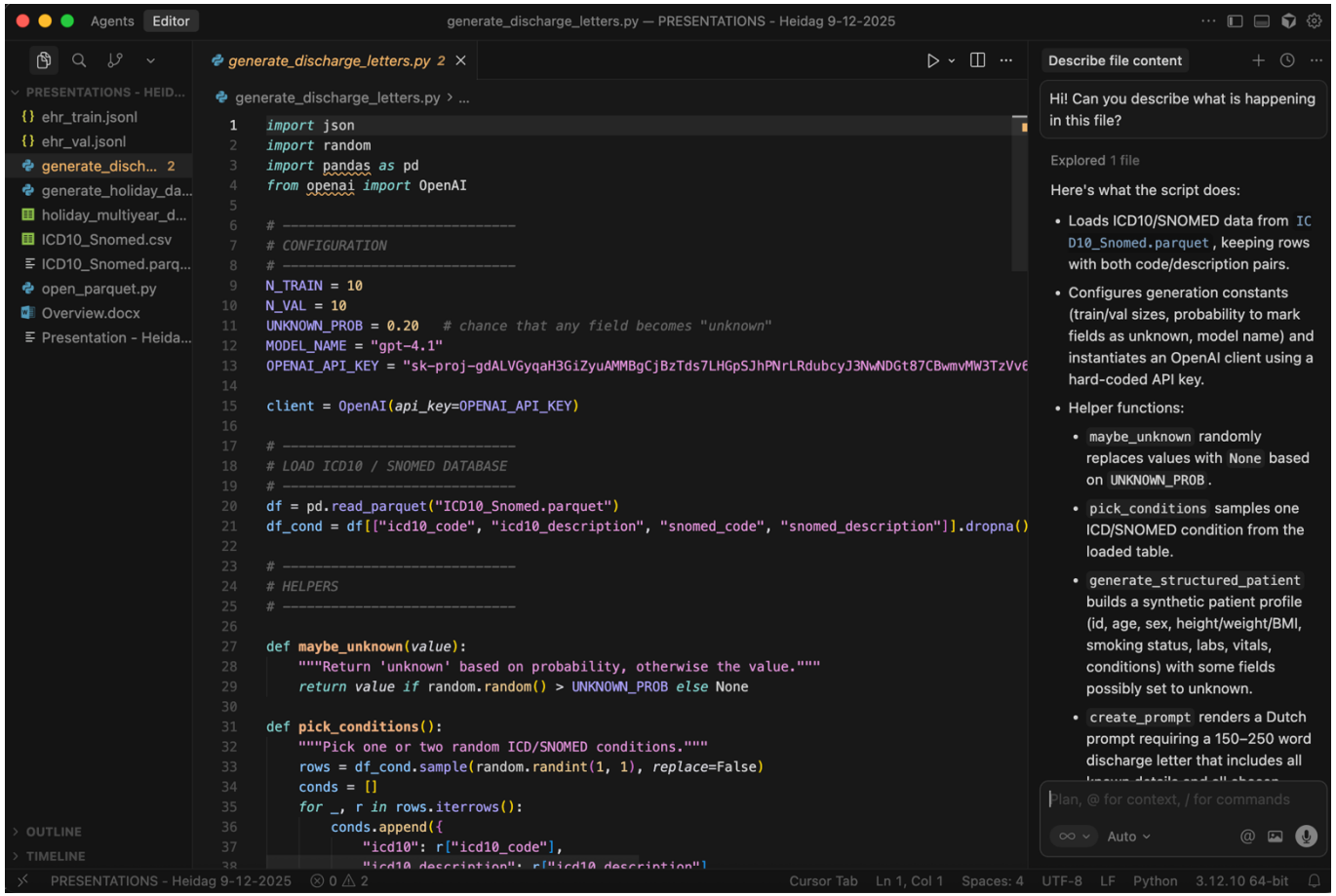
Free text to structured data

Converting structured datasets

Chatbased
- ChatGPT
- **Claude**
- Gemini
- Etc.

Integrated in IDE:
- Github Co-pilot
- GPT Codex
- Windsurf

AI-first IDE:
- Cursor
- …



Cursor

Free text to structured data

Overview

Free text to structured data

Overview

Academic writing assistance

Code generation and translation

Free text to structured data

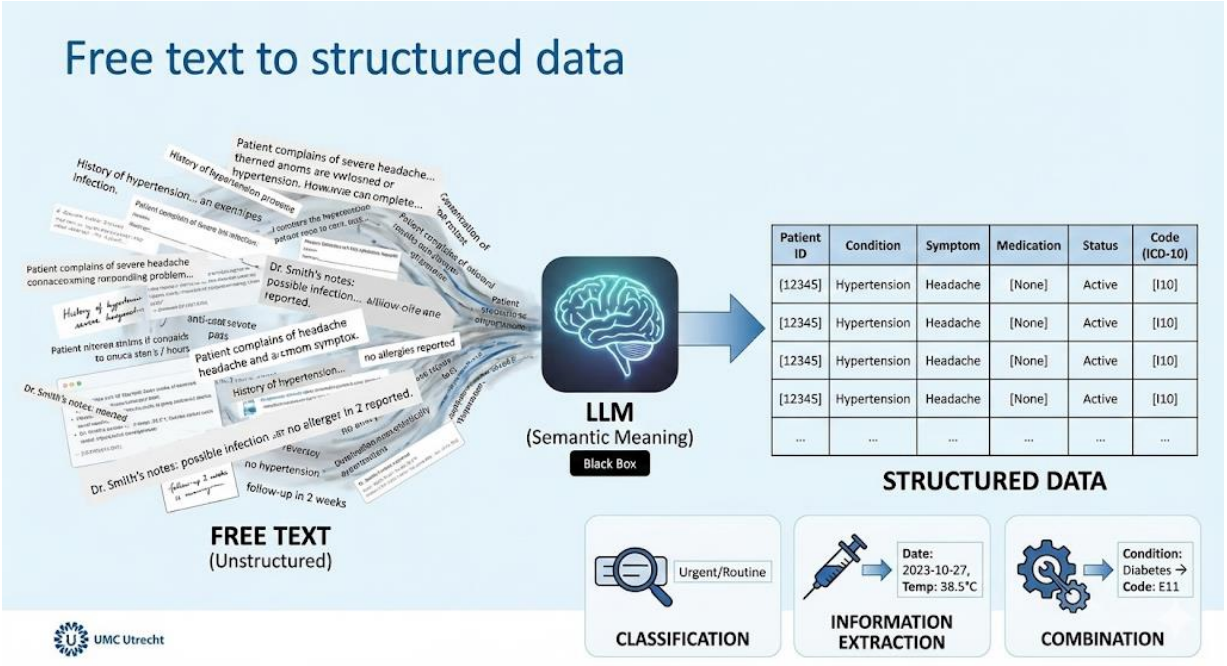
Converting structured datasets

A different type of problem…

- No one-size-fits-all algorithms
- Active field of research
- But LLMs offer a lot of options!

Problem:

- Free text is difficult to structure using ‘rules’
- LLMs can capture ‘semantic meaning’
- But they are a black box
- They do make mistakes!

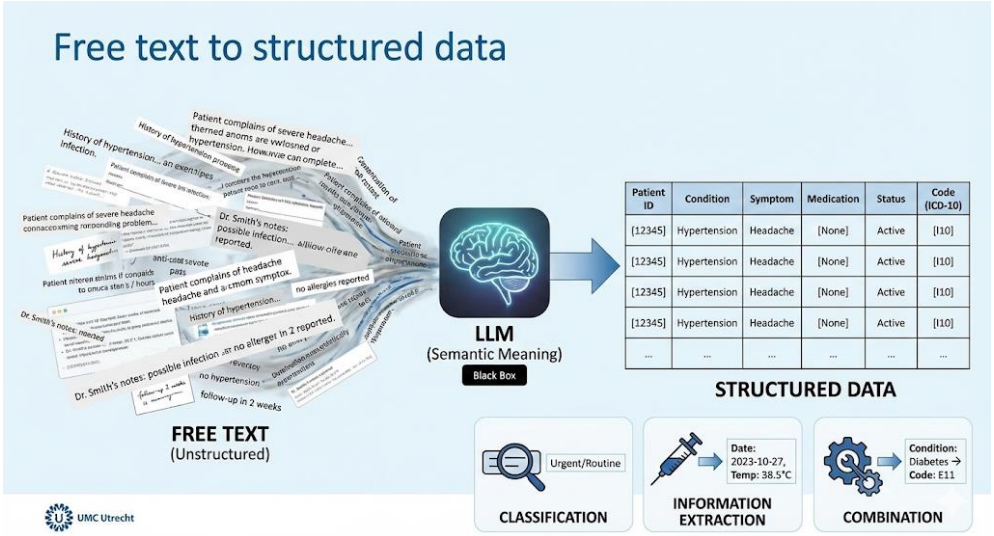


Free text to structured data

Overview

A different type of problem…
- No one-size-fits-all algorithms
- Active field of research
- But LLMs offer a lot of options!

Problem:
- Free text is difficult to structure using ‘rules’
- LLMs can capture ‘semantic meaning’
- But they are a black box
- They do make mistakes!

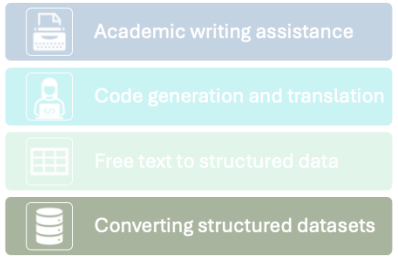


Different approaches:
- In some cases it’s a classification problem, in others information extraction, or a combination!

Examples:
- Interpreting free text fields from tabular data (classification)
- Extracting information in EHRs (information extraction)
- Extracting medical conditions from EHRs and mapping them to clinical classification systems (information extraction and classification)

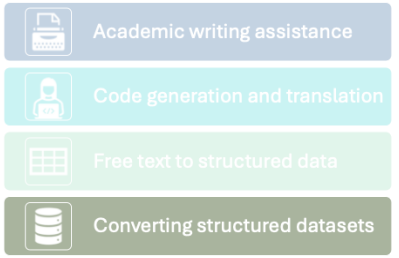
Converting structured datasets

Harmonising (semi-)structured datasets



Converting structured datasets

Harmonising (semi-)structured datasets



Again, a different type of problem…

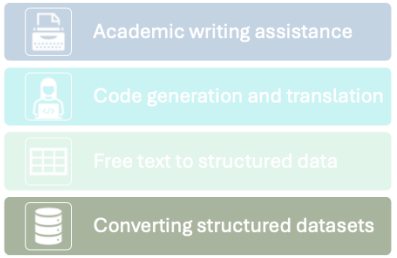
- Structured data appears "clean" but often lacks compatibility
- Traditional approach requires brittle, manual "lookup tables"
- LLMs can act as "semantic bridges" between different standards

Problems:

- Naming can be ambiguous: does "Date" mean admission date or entry date?
- Granularity mismatch: one system might be broad, while the other is specific
- This can cause ”one-to-many" issues: direct 1:1 translations do not always exist
- Hierarchies might not be the same across datasets

Converting structured datasets

Harmonising (semi-)structured datasets



Again, a different type of problem…

- Structured data appears "clean" but often lacks compatibility
- Traditional approach requires brittle, manual "lookup tables"
- LLMs can act as "semantic bridges" between different standards

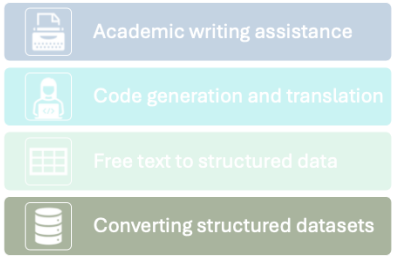
Problems:

- Naming can be ambiguous: does "Date" mean admission date or entry date?
- Granularity mismatch: one system might be broad, while the other is specific
- This can cause ”one-to-many" issues: direct 1:1 translations do not always exist
- Hierarchies might not be the same across datasets

System	Purpose	Scope	Granularity	Typical Use
ICPC	Primary-care	GP encounters	Low detail	GP records
ICD-10	Track diseases worldwide	Broad diseases	Medium detail	Hospitals, billing
SNOMED CT	Standardize clinical terms	Very broad	High detail	Machine-understandable

Converting structured datasets

Harmonising (semi-)structured datasets



Again, a different type of problem…

- Structured data appears "clean" but often lacks compatibility
- Traditional approach requires brittle, manual "lookup tables"
- LLMs can act as "semantic bridges" between different standards

Problems:

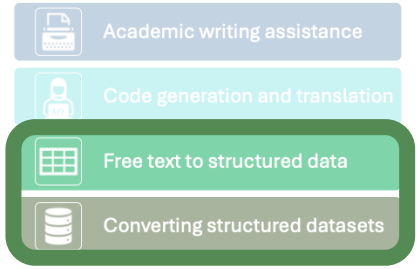
- Naming can be ambiguous: does "Date" mean admission date or entry date?
- Granularity mismatch: one system might be broad, while the other is specific
- This can cause ”one-to-many" issues: direct 1:1 translations do not always exist
- Hierarchies might not be the same across datasets

Examples:

- System migration: mapping fields from one HER to another
- Medical condition classification; translating between different classification systems(e.g., ICD-10 to SNOMED CT)

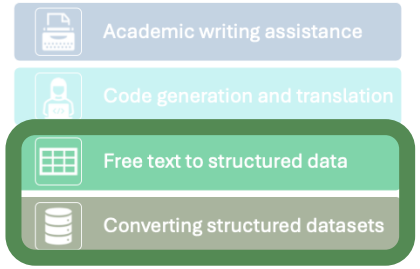
Solutions?

Extracting and harmonising structured data



Solutions?

Extracting and harmonising structured data



Existing (classical) methods

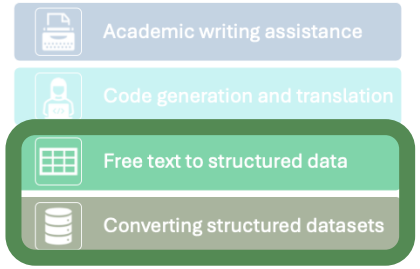
- Official mapping dictionaries (might not cover everything)
- Dutch-specific (ClinNLP): Library of NLP tools for Dutch clinical text
- Proprietary toolboxes:
  - John Snow Labs (no-code Healthcare NLP)
  - AWS HealthScribe
  - Google Cloud Healthcare Natural Language API

Existing LLM methods

- Specialized clinical LLMs (MedGemma, ClinicalBERT)
- Embedding matching for semantic alignment
- …

Solutions?

Extracting and harmonising structured data



Existing (classical) methods

- Official mapping dictionaries (might not cover everything)
- Dutch-specific (ClinNLP): Library of NLP tools for Dutch clinical text
- Proprietary toolboxes:
  - John Snow Labs (no-code Healthcare NLP)
  - AWS HealthScribe
  - Google Cloud Healthcare Natural Language API

Existing LLM methods

- Specialized clinical LLMs (MedGemma, ClinicalBERT)
- Embedding matching for semantic alignment
- …

This is what we will look into during the practical!

Contents

Breakout sessions

Overview

Three concurrent topics:

- Academic writing assistance
- Automated data analysis using code generation
- Free text to structured data

Breakout sessions

Overview

Three concurrent topics:

- Academic writing assistance
- Automated data analysis using code generation
- Free text to structured data

First session:

Choose the session most relevant to your work

Second session (if you want to):

Choose any session you might find interesting

Requirements

- Laptop

Breakout sessions

Overview

Three concurrent topics:

- Academic writing assistance
- Automated data analysis using code generation
- Free text to structured data

First session:
Choose the session most relevant to your work

Second session (if you want to):
Choose any session you might find interesting

Requirements

- Laptop

Competition!
If you are proud of your results, send them to:

r.j.a.kuiper@umcutrecht.nl

and you might win a prize!
(one per category)

Working in groups is allowed, but individual experimentation is encouraged!

Break out session #1

Academic writing assistance

The goal of this practical is to:

- Make you experience what using LLMs for writing is like
- Show the current capabilities
- Show the current shortcomings
- Help you recognize AI writing when reviewing!

The goal of this practical is not to make you use LLMs for academic writing!

Assignment:

Write a short literature review on the following topic:

"The effect of the (Christmas) holiday periods on mental-health outcomes"

Break out session #1

Academic writing assistance

Topic (you can use this to prompt the AI):
"Please provide me with a full literature review that could be published in The Lancet on the effect of the (Christmas) holiday periods on mental health outcomes. Only use peer-reviewed academic sources, and give one or multiple references for each statement. Do not use any embellishing words."

Try to use a fully automatic workflow using one of these:
jenni.ai - consensus.app – perplexity.ai –
chatgpt.com – gemini.google.com – Asreview

For some models, a dataset is necessary. A curated relevant(?) dataset is available, alongside more information, in the folder in Teams

Break out session #2

Automated data analysis using code generation

The goal of this practical is not to make you use AI coding assistants.
The goal is rather to:

- Let you experience AI-assisted coding
- Show current strengths and weaknesses
- Explore differences between coding assistants
- Check out whether AI might help in your workflow

Assignment:
Write a small script using an AI coding assistant to analyse and visualize holiday-themed dataset.

Download the following AI coding assistant:
Cursor for Python, or Positron for R

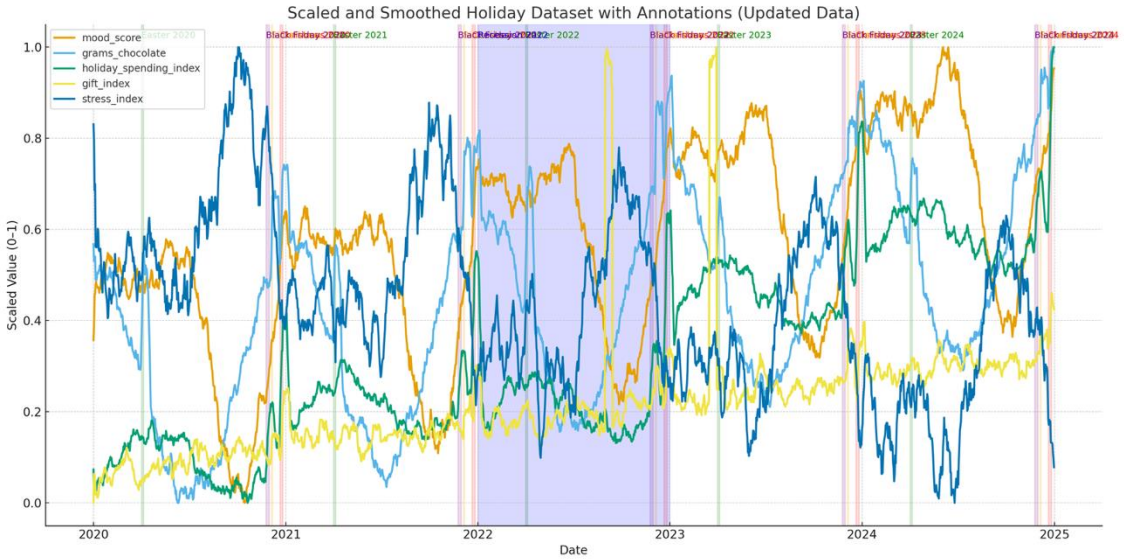
Folder in Teams:
Breakout #2 - Data analysis with AI code generation

Break out session #2

Automated data analysis using code generation

The data contains 5 variables for 100 subjects over 5 years:
holiday_multiyear_dataset_multisubject.csv

- **mood_score:** Daily mood score
- **grams_chocolate:** Chocolate consumption
- **holiday_spending_index:** Euros spend on gifts
- **gift_index:** Number of gifts received
- **stress_index:** Stress

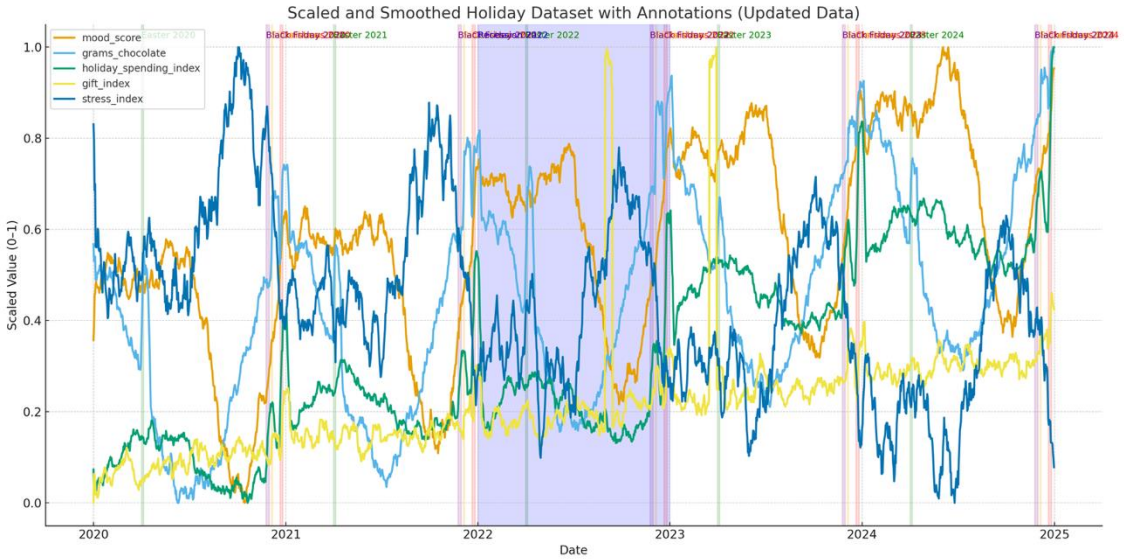


Break out session #2

Automated data analysis using code generation

The data contains 5 variables for 100 subjects over 5 years:
holiday_multiyear_dataset_multisubject.csv

- **mood_score:** Daily mood score
- **grams_chocolate:** Chocolate consumption
- **holiday_spending_index:** Euros spend on gifts
- **gift_index:** Number of gifts received
- **stress_index:** Stress



Depending on your expertise, try to answer some of the following questions:

Try to find interesting patterns in the data:

1. Clean missing data and correct outliers.
2. Identify seasonal patterns across years.
3. Quantify holiday effects (Sinterklaas, Christmas, Easter, Black Friday).
4. Compute correlations between variables.
5. Explore causal relationships (e.g., mood ↔ stress ↔ chocolate).
6. Train a predictive model.
7. Test causal hypotheses (e.g., does chocolate cause mood changes?).
8. Create an insightful or humorous visualisation of the patterns.

Break out session #3

Free text to structured data (information extraction)

The goal of this practical is to:

- Let you work with LLMs for clinical information extraction
- Show you how to load, run and finetune small LLMs
- Reveal typical difficulties with information extraction using LLMs
- Make you think about LLM data extraction pipelines!

This is not a full, working information extraction pipeline!

Assignment:

- Use the provided python notebook.
- Read the instructions for each step and run the code blocks.
- Inspect the outputs to see if you understand what's going on.
- Compare generic LLM performance vs. finetuning and embedding-based matching.

Link:

Breakout #3 - Free text to structured data using AI

Break out session #3

Free text to structured data (information extraction)

Tips:

- Use Google Colab to run the notebook (**EHR_to_structure.ipynb**) so you can use a (free!) GPU
- Upload all the relevant files to the environment or folder where you run the notebook.
- Make sure to change the runtime type using the upper menu bar:
  Runtime -> Change Runtime Type -> Choose “T4 GPU”
- Do not change the model type at first, but first make sure you can run the whole notebook.

Contents