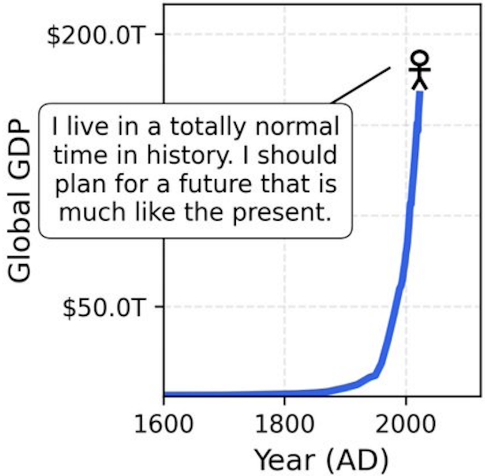
AI Literacy for Studying, Research, and Life

How LLMs Work, Where Bias Comes From, and What You Can Do With Them

Dr. Matt DiGiuseppe
Associate Professor
Institute of Political Science
Leiden University

March 2026



[h/t Tim Urban, Wait But Why]

Today's Plan

1. **How LLMs actually work** — tokenization, prediction, training
2. **Where bias comes from** — and why it matters for your work
3. **What you can do with them** — research applications
4. **AI for studying** — learning better (and what not to do)
5. **Practical tools** — prompting, coding, and the real product
6. **AI for life** — the job market and why this matters now

At its core, an LLM does one thing:

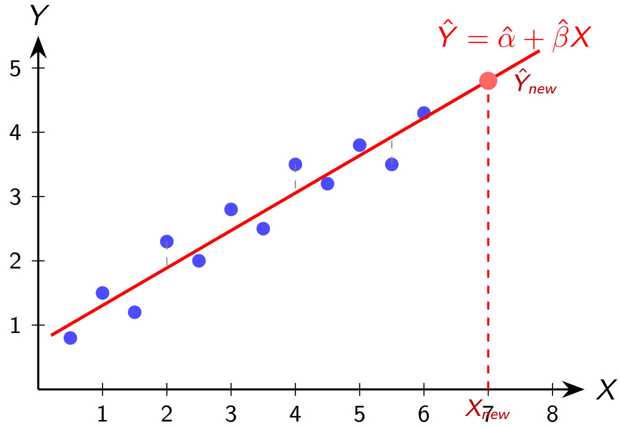
Predict the next word.

Word	Probability
ability	0.002
bag	0.071
box	**0.085**
...	...
zebra	0.001

Output

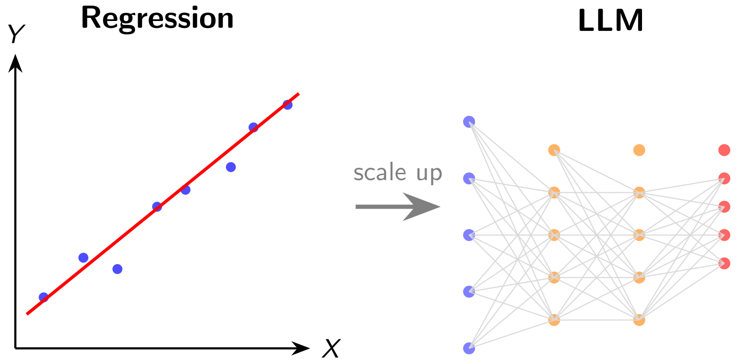
Given all the words so far, what word comes next? Do this one token at a time → paragraphs, essays, conversations.

Remember Regression?



Data → fit a line → minimise prediction errors (dashed gaps).
The **slope** and **intercept** are *parameters* learned from data.

Regression → LLM: Same Logic, Different Scale



2 parameters (α, β)
Minimise squared errors

Billions of parameters
Minimise next-word prediction error

Same principle: adjust parameters to reduce prediction error.
A regression learns 2 numbers. An LLM learns *billions* from the entire internet.

Why This Analogy Matters

Training data shapes behavior
Biased or narrow data produces biased or narrow outputs

Generalisation matters
Good models learn patterns not just memorised examples

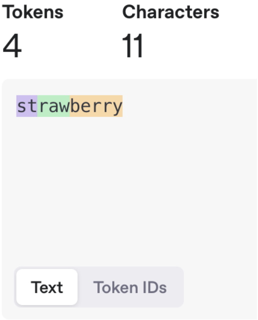
Scale changes capabilities
More data + parameters can unlock new behaviors

Prediction can build rich representations
Optimising next-token prediction can still produce reasoning-like abilities

If you understand regression, you already understand the core logic of LLMs: fit parameters to reduce error, then ask whether the model *generalises*. What changes is not the principle, but the scale and the complexity of what gets learned.

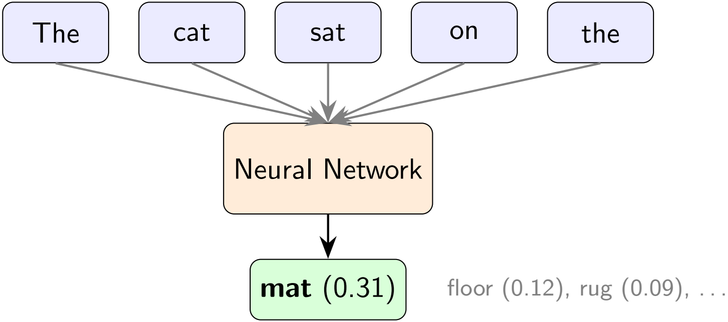
Tokenization

LLMs don't read words — they read **tokens** (subword chunks).



- "strawberry" → tokens: `st` — `raw` — `berry` — the model never sees individual letters.
- This is why earlier models couldn't count R's in "strawberry." (Newer reasoning models work around this by spelling step by step.)

Next-Token Prediction in Action



The model assigns a **probability** to every token in its vocabulary, then samples.

Temperature controls how "creative" vs. "safe" the output is.

How Is an LLM Trained?

Pre-training
Learn language from the internet

→

Fine-tuning (SFT)
Learn to chat from human examples

→

RLHF
Learn to be helpful & safe

Trillions of tokens
Billions of params
$100Ms–$Bs in compute

Human-written Q&A
"Text completer"
→ "Chatbot"

Human raters rank outputs
Reward model trained
LLM tuned to maximise score

Each stage shapes the model differently — and introduces different **sources of bias**.

Pre-training: Learning Language from the Internet

Self-supervised learning: mask a word, predict it, adjust parameters. Repeat trillions of times.

Data
Common Crawl, books, Wikipedia, code, forums

Scale
Trillions of tokens
Billions of parameters
\$100Ms–\$Bs in compute

Bias enters here
$\sim$90% English
Internet $\neq$ reality
Over-represents Western, male, young, online

Result
A "text completer"
No chat ability yet
Encyclopaedic but raw

The largest source of bias by volume. The model's world-view is set here — everything after is fine-tuning at the margins.

Fine-tuning (SFT): Teaching It to Chat

Human annotators write **example conversations** — the model learns to imitate them.

User: What caused the 2008 financial crisis?

Assistant: The crisis had multiple causes, including subprime lending...

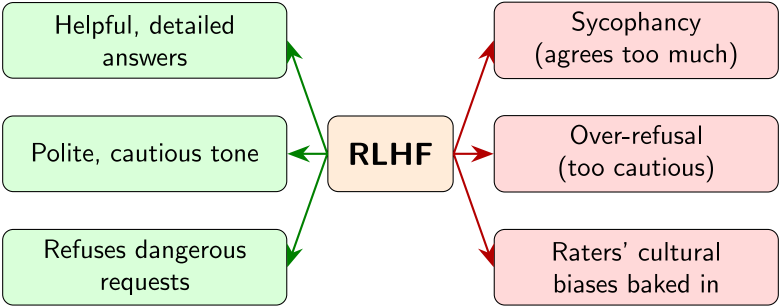
↓

$\times$ thousands of examples $\rightarrow$ "text completer" becomes "chatbot"

Where does bias enter?

- Annotators decide what counts as a "good" answer — tone, length, hedging, refusals.
- Small team, narrow demographics $\rightarrow$ their values become the model's defaults.
- Less data than pre-training, but high leverage: it shapes the *personality*.

What RLHF Does (and Distorts)



Raters are typically English-speaking knowledge workers. Their judgements about "helpful" and "harmful" become the model's defaults.

The Hidden Layer: System Prompts

After RLHF, companies add **hidden instructions** the user never sees.

```
You are ChatGPT. Be helpful, harmless, and honest.
Do not discuss competitors. Do not generate code for
weapons. Always respond in the user's language...
```

- Shape what the model says *and refuses to say* — refusals, tone, personality.
- The same base model behaves differently in ChatGPT, Copilot, and Claude because of different system prompts.
- Companies use them to enforce brand, legal compliance, and content policy.
- **You can set your own** in Projects and Custom Instructions — this is where the "easy wins" live.

Where Does Bias Come From?

Pre-training → **Fine-tuning** → **RLHF**

$\sim$90% English
Western, male, young
internet $\neq$ reality

Example writers'
values shape
"good" answers

Raters' preferences
become the model's
preferences

These biases are **systematic, not random** — they don't cancel out.

Like running a regression on a convenience sample:
your estimates reflect *who*
showed up, not the population.

Why This Matters for Research

Text classification
Training shapes how it reads ideology & sentiment

Synthetic subjects
"Opinions" reflect RLHF, not real populations

RAG / deep research
Hallucinations more confident where data was sparse -getting rarer

Non-Western contexts
Worst performance where training data is thinnest

Treat LLM output like data from any source.
Validate, cross-check, report limitations.

What Can LLMs Do for Your Research?

Classify text
zero-shot, no training data

Create latent variables
scaled comparisons

Deep-dive literature
RAG, NotebookLM

Qualitative research
search, read, synthesise

Simulate subjects
synthetic surveys & experts

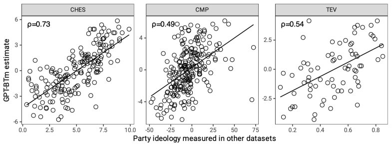
Write & debug code
vibe coding, agents

Always validate against human judgement. Tools, not oracles.

Latent Variable Construction

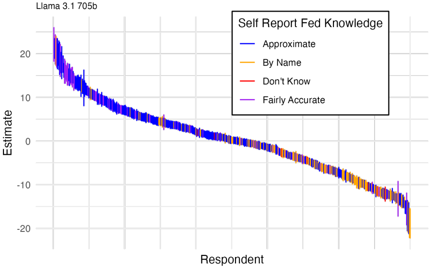
LLM pairwise comparisons $\times$ thousands $\rightarrow$ estimate relative positions.
Works with party labels *or* open-ended text.

Figure 1: Benchmarking European Parties' Left-right Positions according to GPT-3.5 against experts, manifestos and opinion polls (Reference Year: 2009).



Notes: In each facet, we plot the left-right ideological positions of European parties obtained applying a Bradley-Terry model to the pairwise comparisons performed by GPT-3.5 (GPT-BTm), in reference year 2009. In each panel we employ different validation datasets, namely: Chapel Hill Expert Survey (CHES), Comparative Manifesto Project (CMP), True European Voter (TEV).

Di Leo et al. 2024



DiGiuseppe & Flynn 2025

RAG and Qualitative Research

RAG — upload documents, query with natural language.

- Ask questions of a corpus
- Summarise hundreds of articles
- Create a podcast from a paper

NotebookLM — ChatGPT Projects — Claude Projects

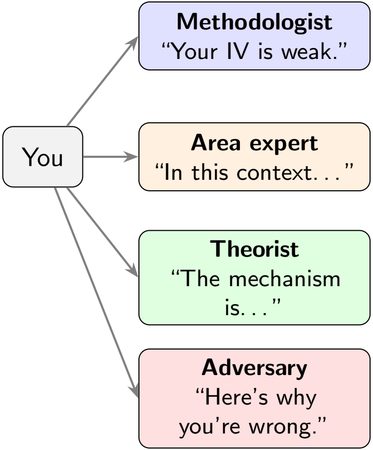
Deep Research agents — search, read, synthesise autonomously.

- Prompt template + execution matrix
- Batch via API → cross-compare
- Spot-check citations (they hallucinate)

LLM output = first draft from a well-read but fallible RA.

Simulating Subjects and a Panel of Experts

- **Synthetic respondents:** LLMs adopt personas, take surveys. Exploratory, not confirmatory.
- **Generative agent societies:** LLM agents that remember, reflect, plan $\rightarrow$ emergent social behaviour (Park et al. 2023).
- **A pool of experts on demand:** methodologist, area specialist, theorist — all at your fingertips. Not sure about a design choice? Have an adversarial AI argue the other side.



AI as a Study Partner

Tutor mode
"Explain [concept], then quiz me. Start easy, get harder."

Custom quiz bot
Upload slides to a Project. "Be a strict exam coach."

Visualise concepts
"Create an interactive illustration of adding controls in a regression "

Debate partner
"Argue the other side. Push back on my weak points."

Use Voice Mode for Quizzes and Feedback

Offload the Search, Not the Thinking

Offload this

- Searching & filtering literature
- Generating diagrams to grasp a concept
- Getting a concept explained a new way
- Reformatting, cleaning, admin

Don't offload this

- Reading the summary *instead of* the source
- Submitting AI text as your own
- Accepting answers without checking
- Avoiding the struggle of learning

Use AI to study harder, not to study less.

Easy Wins: Five-Minute Setup, Permanent Payoff

Kill the sycophancy
Base instructions: "Be direct. Push back when I'm wrong."

Teach it your style
Feed it your writing, ask it to create a style profile or skill

Set your level & context
"I'm a Master's student in poli sci at a Dutch university."

Pin your research topic
Add your thesis question to Projects / Custom Instructions

Use Memory
Let is get to know you to auto load context

These go in **base settings** (ChatGPT Custom Instructions, Claude Projects, or system prompts). They persist across every conversation — set once, benefit forever.

Prompting and Context

Give it a role
"You are a methods expert reviewing my thesis draft."

Share files
Upload data, codebooks, papers directly.

Use Projects
Pin instructions that persist across conversations.

Be explicit
State variable names, theory, constraints.

Think like a manager: delegate clearly, provide context, check the work.

Don't Use the Free Version!

The free tier is a **demo**, not the real product.

Free tier
Older, smaller models
Limited context & tools

→

Paid (€20/mo)
Frontier models
File uploads, web, code

→

API
Cheapest at scale
5,000 texts ≈ €2–5

You wouldn't judge Netflix by the ad tier with 480p.
The paid tier costs less than your coffee habit.

Vibe Coding and Beyond

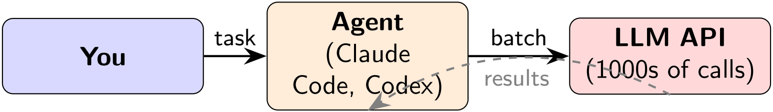
"English is the hottest new programming language." — Andrej Karpathy

Describe what you want; the LLM writes the code.

Chat-based	ChatGPT, Claude, Gemini, DeepSeek
AI IDEs	Cursor, Windsurf, Kiro (Amazon), Positron (R/Python)
Cloud agents	Codex (OpenAI), Claude Code, GitHub Copilot Agent Mode
Local / private	Ollama — free, no data sharing, runs on your laptop

Tell the LLM you're a novice. Break tasks into parts. Share error messages. Iterate.

AI Using AI: Agents + API



Example 1: "Classify the sentiment of 5,000 parliamentary speeches." The agent writes code, sends API calls, saves a clean dataset. Cost: a few euros. Time: overnight.

Example 2: "Here are all the materials from my International Development course and an API key. Build me an interactive quiz app. Add a chatbot that explains what I got wrong. Keep testing me on my weak spots." The agent builds the whole thing — frontend, logic, API calls — from a single prompt.

AI orchestrating AI — this is how you build tools, not just get answers.

LLM Development Environments (LDEs)

Claude Code and **Codex** aren't just for coding — they're general-purpose work environments.

- Give it 50 PDFs → extract, compare, and summarise across them.
- "Run these regressions, make a table, write up the results."
- "Check every citation in my thesis against the bibliography."

A capable RA with a terminal. You describe the task; it figures out how.

No one knows what AI will do to jobs in the long run.

In the short run, we have a few clues.

The Job Market Is Shifting Under Your Feet

5.3% graduate unemployment UK 2025 — highest since 2016

~**50%** drop in entry-level tech hiring since 2022

Junior analyst, paralegal, copy editor — roles AI automates **first**

Firms skip junior hires, give AI tools to senior staff instead

The bottom rungs of the career ladder are disappearing.
AI isn't the only cause — but it's accelerating the trend.

What This Means for You

Disappearing	Growing
Summarising reports	Evaluating AI output critically
Basic data entry & analysis	Client relationships & judgement
First-draft writing	Designing research
Routine research tasks	**AI orchestration**

Universities teach the left column. You need to teach yourself the right.

How to Stand Out

- **Build a portfolio, not just a CV.** Dashboards, scraped datasets, policy briefs, apps — show what you can *do*.

How to Stand Out

- **Build a portfolio, not just a CV.** Dashboards, scraped datasets, policy briefs, apps — show what you can *do*.
- **Work with AI, not just use it.** Prompting, validating, chaining tools — this is a professional skill.

How to Stand Out

- **Build a portfolio, not just a CV.** Dashboards, scraped datasets, policy briefs, apps — show what you can *do*.
- **Work with AI, not just use it.** Prompting, validating, chaining tools — this is a professional skill.
- **Use AI for the job search.** Job ad + CV $\rightarrow$ tailored cover letter. Company report $\rightarrow$ interview prep. Simulate tough interviews — USE YOUR OWN VOICE!

How to Stand Out

- **Build a portfolio, not just a CV.** Dashboards, scraped datasets, policy briefs, apps — show what you can *do*.
- **Work with AI, not just use it.** Prompting, validating, chaining tools — this is a professional skill.
- **Use AI for the job search.** Job ad + CV → tailored cover letter. Company report → interview prep. Simulate tough interviews — USE YOUR OWN VOICE!
- **Develop durable skills.** Critical thinking, communication, domain expertise — these *appreciate* as AI improves.

How to Stand Out

- **Build a portfolio, not just a CV.** Dashboards, scraped datasets, policy briefs, apps — show what you can *do*.
- **Work with AI, not just use it.** Prompting, validating, chaining tools — this is a professional skill.
- **Use AI for the job search.** Job ad + CV → tailored cover letter. Company report → interview prep. Simulate tough interviews — USE YOUR OWN VOICE!
- **Develop durable skills.** Critical thinking, communication, domain expertise — these *appreciate* as AI improves.
- **A recruiter does not care about your principled stand against AI.** They care whether you can do the job. The candidate who can will get it.

The Real Career Skill

Your future boss is Gen X.

They grew up without this technology.
Many are still figuring out what AI can do.

You can do in hours
what they hire teams to do in weeks.

- Summarise 200-page reports in minutes.
- Build dashboards and prototype tools without a developer.
- Analyse qualitative data that would take a team of interns a month.

But Only If You Know How It Works

Anyone can type a question into ChatGPT.

The people who stand out **understand the tool** —
its strengths, its limits, and its biases.

- Know when to trust it and when to check.
- Know what "RLHF bias" means when the AI gives a weird answer.
- Know when to use the API instead of clicking in a chat window.

That's AI literacy. And it's a career advantage right now.

Key Takeaways

1. LLMs are **next-token prediction machines** — regression with billions of parameters.
2. The training pipeline introduces **systematic biases** at every stage.
3. Powerful for research, studying, and work — but **always validate**.
4. **Understanding the tool** is what separates casual users from effective ones.

Questions?

m.a.di.giuseppe@fsw.leidenuniv.nl