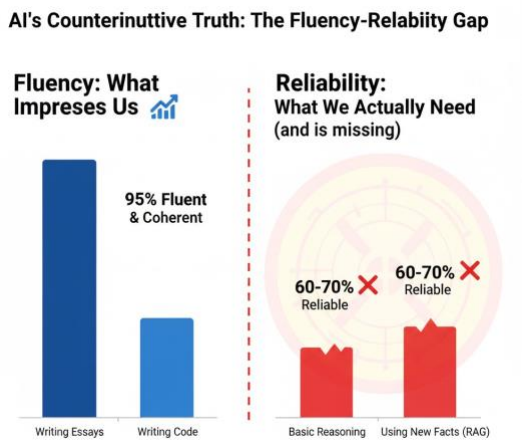
Myths and Merits of LLMs

It presents a deeply counterintuitive and critical insight for any leader implementing AI.

The central, counterintuitive finding is this: We are focusing on the wrong things.

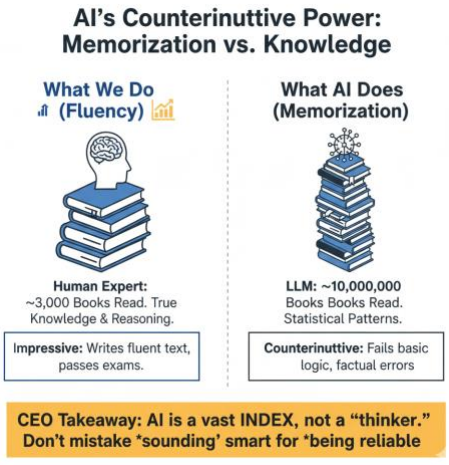


We are impressed by an LLM's ability to perform tasks we (humans) find *hard* (like writing fluent essays or code), but we are blind to its critical failures in tasks we find *easy* (like basic reasoning, simple math, or reliably using new information).

LLMs are not "reasoning" at all. They are "Thinking Fast"—using massive memorization to predict the next word—when what businesses need is "Thinking Slow"—the ability to identify structure, follow constraints, and reason consistently

This memo summarizes the research, its most memorable examples, and the critical takeaways for a leadership audience.

Lesson 1: Fluency Is Not Intelligence (The "10 Million Book" Analogy)



We, as humans, are trained to equate linguistic coherence with intelligence.

this is now a critical error

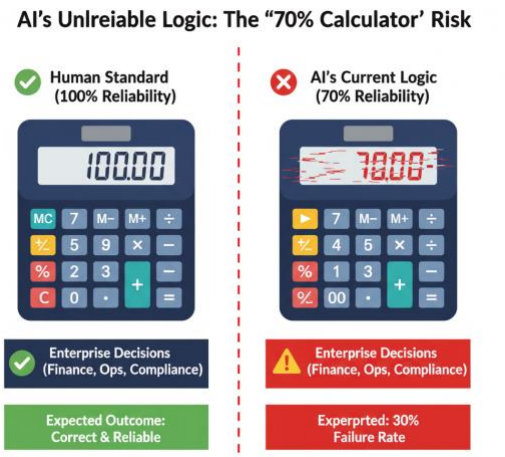
- **The Analogy:** An avid human will read, at most, **3,000 books** in their lifetime. The LLMs you use today have been trained on the equivalent of **10 to 100 *million* books**
- **The Insight:** The model is not "smart." It is an *index* of everything humanity has ever written. It can retrieve and re-pattern this information fluently, but it does not *understand* it.

CEO Takeaway: You are not deploying a "thinker." You are deploying a system with near-perfect memorization but flawed reasoning.

Myth 1: AI Can "Reason" (The 70% Calculator)

The most dangerous myth is that LLMs can perform reliable, logical decision-making. The research shows they cannot—they merely *mimic* reasoning they have seen in their training data.

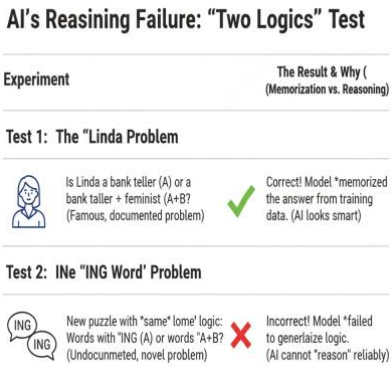
- **The Analogy:** "Imagine that your calculator would give you the answer correctly 70% of the time". This is the state of LLM reasoning today. While 60-70% accuracy is impressive for summarization, it is catastrophic for enterprise decision-making (e.g., in finance, logistics, or compliance), where 100% accuracy is the standard



Replicated Analysis: The Logic Failure

This experiment shows how LLMs *fake* reasoning.

Experiment	The Problem	The Result & (Why)
Test 1: The "Linda Problem" - https://en.wikipedia.org/wiki/Conjunction_fallacy	A famous psychological puzzle ("Is Linda *A* or *A+B*?"). It is famously documented all over the web	**Correct** (The model *memorized* the answer from its training data)
Test 2: The "ING Word" Problem *"What is larger? The number of 7 letter english words with the character ´n´in the 6th position OR the number of 7 letter english words that end with the characters ing´?*	A new puzzle with the *exact same logical structure* as the Linda problem, but using different words (words with 'n' vs. 'ing')	**Incorrect** (The model *could not generalize* the logic to a new problem it hadn't seen before)



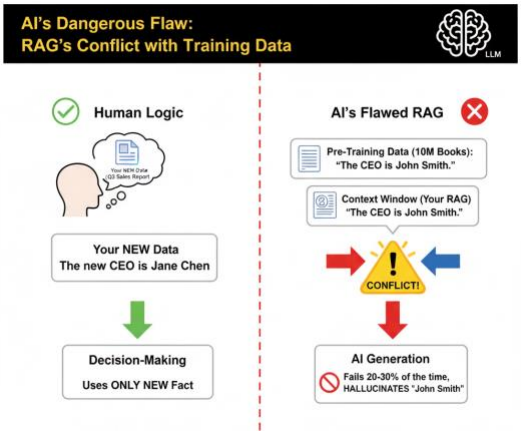
Replicated Analysis: The Optimization Failure

The research tested LLMs on solving complex constraint optimization problems (e.g., "what's the most efficient shipping route?").

Approach	The Result & (Why)
Approach 1: "LLM as Reasoner" Ask the LLM to *solve* the problem directly.	**Fails** (Accuracy is "pretty bad... doesn't solve almost any of the problems")
Approach 2: "LLM as UI" Ask the LLM to *translate* the problem into code for a classic, reliable solver (like an ILP engine)	**Succeeds** (Accuracy is "pretty good." The LLM acts as a universal *translator*, not the *thinker*).

CEO Takeaway: Do not use LLMs to *make* decisions. **Use LLMs as a universal UI** that translates human language into commands for your *existing*, reliable tools (e.g., calculators, databases, and solvers).

Myth 2: AI Can "Find" Your Data (The 'R' in RAG is Broken)



The entire industry is betting on Retrieval-Augmented Generation (RAG)—the idea that you can just "point" an LLM at your company's data (reports, tables, etc.) and it will use it. The research shows this is dangerously flawed.

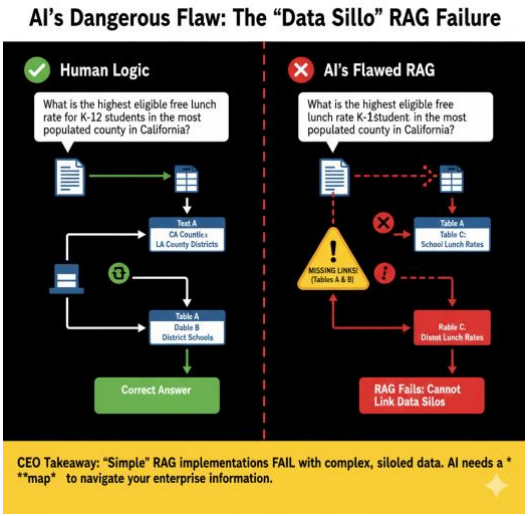
Replicated Analysis 1: The "Context vs. Training" Conflict

LLMs struggle to prioritize new information (your data) over their old training (the 10M books).The question is ***when did Athens emerge as an economic power?***

- **The Experiment:** The model is given a new fact in context (e.g., "Athens emerged in the **4th century**"). The model *knows* from its training that the answer is the "6th century"
- **The Result:** In 20-30% of cases, the model *ignores* the new context you provided and *hallucinates* the old answer from its training

CEO Takeaway: When your enterprise data changes (e.g., new sales figures, new CEO), your RAG system will *fight* that new information, leading to hallucinations and bad decisions.

Replicated Analysis 2: The "Data Silo" Failure



Your data isn't in one document. It's in multiple, heterogeneous tables and texts. LLMs cannot navigate this complexity.

- **The Query:** "What is the highest eligible free [lunch] rate for K-12 students in schools in the most populated county in California?"
- **How a Human Solves It:**
  1. Find the most populated county (e.g., LA County) from *Text A*.
  2. Find all school districts in that county from *Table A*.
  3. Find all schools in those districts from *Table B*.
  4. Look up the rates for those schools in *Table C* and find the max.
- **How Standard RAG Solves It: It fails.** It finds *Text A* (about counties) and *Table C* (about rates), but it cannot *semantically link* them. It doesn't know it needs to find the "bridge" tables (A and B) in between

CEO Takeaway: A "simple" RAG implementation will fail. The AI's weakness is not generation; it is *reliable retrieval across multiple sources*.

The Path Forward: Your 2-Step Action Plan

1. Stop Thinking "One Big Brain." Start Thinking "Modular Systems."
   The future is not a single, all-knowing LLM. It is a modular system [15:24] where the LLM acts as an orchestrator that delegates tasks to specialized solvers:
   - For **math**, it calls a calculator.
   - For **logic**, it calls a solver.
   - For **data**, it calls a database query.
2. Invest in Your Data Structure, Not Just the AI.
   The solution to the RAG/retrieval problem is not a "better" LLM. It is better data enrichment.

   The research shows that by offline processing—using LLMs to pre-build better indices, summaries, and links for your data —you can dramatically improve retrieval accuracy (e.g., boosting a document from 67th place to 7th).

   Your investment should be in making your data "findable" for the AI.