Evaluation Console

lm

6

Run ID	Chunk Size	Context Size	Temperature			
	600	240				
1004						
1003						
1002	400			Mistral 7B		
1000	200					
			0	GPT	0.2	
999		pragn	natic ap	oproac	ch to	
	⊖ Status III Columns Ξ De	Buildir		rativo		
ホ RUN ID	ೆ EXPERIMENT NAME	DUIIQII		auve	8 JUNTION	
001005 🔘	Data Extraction					
001004 🗊	Data Extraction	O Complete	nnlicati	ions		
001003 🔘		◎ Complete	ppnoau			
001002 🛛						
001001 😡						
001000 🔘						
000999 🛛						
000998 🕔						
000997 0		Failed				
000996		⊙ Comple	CTMAI			
000995 🗊		⊙ Comple				
000994		O Complete	0.59	0.82		
000993 🖪						

Agenda

- 1. About Me
- 2. Not every problem needs a Gen Al solution
- 3. Hype- & jargon-free patterns for building AI applications
- 4. The Last Mile problem of Evaluation
- 5. Customary plug of LastMile Al
- 6. Q&A

About Me

What we do at LastMile AI

Team mission:

Enable <u>software engineers</u>, not just ML research scientists, to ship generative AI applications <u>with confidence in production</u>

About the speaker:

- CEO of LastMile Al
- Building developer tools my entire career VS Code and build systems at Microsoft, Jupyter notebook platform at Meta
- Helped build AI infra for Meta's ML engineers and data scientists.



Sarmad Qadri





There is a lot of hype around generative AI

...but not every problem needs a Gen Al solution

Al innovation is accelerating:

- Models keep getting better and smaller
- Cost/token down 10x-100x

But we are at the risk of seeing every problem through the lens of Gen Al, whether it deserves to or not.

- Not every problem is an Al problem
 - Example: Chat interfaces everywhere
- Not every Al problem is a Gen Al problem
 - Example: RAG vs. IR or RecSys



Step-by-step guide to building AI applications That Work[™]

Hype- & jargon-free patterns that can help you scale

It takes days to build a prototype

Gen Al Prototype
User Interface
Prompts
Hosted LLM



Messy development loop

But it takes months to make the prototype **production ready**



Step 1: Is this a hammer looking for a nail?

Will your problem benefit from an AI solution?

- A. Clearly define the business problem you are trying to solve.
- B. Identify the best implementation for your problem
 - *Example*: For math, use a calculator. For fixed patterns, a regexp will do.
- C. If the simplest approach still requires a generative model, then proceed to Step 2.
 - *Example*: Intent recognition, natural language processing, information synthesis



Step 2: Cut through the hype and jargon

There are a lot of distractions with the Shiny New Thing

Last year everyone was talking about vector databases, then RAG, then prompt optimization frameworks, and now we're talking about agentic workflows.

We don't need to reinvent everything:

- "LLM Observability" is just... observability.
- "Multi-agent workflows with memory" ~= Workflow orchestration with persistent state
- "Agent" ~= LLM with tool use (in vast majority of cases)
- "Prompt management" ~= just use source control



Step 3: Understand the limits of the current SOTA

Stay within the limits to ensure a robust application experience

tl;dr:

- Really good for enhancing retrieval applications
- Not yet great for <u>unconstrained</u> agentic workflows

Tips:

- A. Almost every use case in enterprise boils down to information retrieval, extraction, synthesis.
- B. For agentic workflows, make it more deterministic by defining a state machine of interactions
 - Example: a codemod agent workflow can use its domain knowledge to define a state machine DFA



Step 4: Build the system!

Don't forget about machine learning pre-ChatGPT

- Avoid unnecessary frameworks, except to prototype quickly.
 - As you iterate on the system, the abstractions often get in the way.
- For retrieval systems, don't forget about IR research from pre-LLM days:
 - **BERTopic** for topic extraction
 - RBAC and data refresh/indexing
- Enhance experience with fine-tuned LLM's and rerankers
 - Example: Incident response (Meta): 42% accuracy in incident RCA (root-cause analysis)



Step 5: Set up the harness around your application

Guardrails, monitoring & observability, etc.

Embrace the complexity, without overcomplicating things.

Al systems are distributed systems, and require a lot of the same primitives:

- Guardrails to constrain behavior (more on this later)
- Observability
- Feedback loop to improve the system (including data for fine-tuning)



Source: Building a generative Al platform (Chip Huyen)

Step 0: The Last Mile problem of Evaluation

Define how you are going to evaluate/measure performance

Evaluation: how do I know my application is performing well?

Al evaluation breaks the traditional SDLC:

• Al applications introduce non-determinism, which is different from standard integration tests/unit tests.

Current state-of-the-art for evaluating AI systems is LLM-as-a-judge. This approach is:

- Expensive
- Unreliable (LLMs weren't designed to be evaluators)
- Hard to customize for your specific application





Custom evaluator models for evaluation, testing and guardrails of AI applications

AutoEval

State-of-the-art evaluator models for evaluating and testing LLM & RAG applications. More performant and efficient than LLM-as-a-judge techniques.

	HaluEval	WikiEval
SotA Baseline	85%	85%
LM P(Faithful)	86%	98%

Supported Evaluators: Faithfulness, Correctness, Toxicity, and Relevancy

Fine-Tuning Evaluators: Cost-efficient enough to be fine-tuned to your business use case.

Guardrails: 100x faster than LLM-as-a-judge allows it to be used during online inference.

Model Console > Øp-Faithful			E.		
© p-Faithful Evaluation Model • B J.4 12th • 9 35 Balance D Overview O History					
7d 24h 30m <4 All Requests	-	▷ Try p-Faithful The model is currently online, see what it can do			
Live metrics showcasing model health and performance		ІЛРИТ РИСМРТ	CONTEXT		
	4.91	OUTPUT			
19.00.00 19.00.00.01 19.00.00.02 19.00.00.03 19.00.00.004	19:00:00.005 19:00:00.006				
Latency	522.8 ms	K Latest Queries			
		List of recent live requests and associated p-faithful score INPUT: What's the capital of France? Output: Paris	0.		
19:00:00 19:00:00.001 19:00:00.002 19:00:00.003 19:00:00.004	19:00:00.005 19:00:00.006	INPUT: How many legs does a spider have? Output: Eight	0.		
Total Tokens	151 token	INPUT: Who wrote "Romeo and Juliet"? Output: William Shakespeare	0.		
		Input: What's the chemical symbol for gold? Output: Bh	Θ.		
7		Input: In which year did World War II end? Output: 1945	0.		
19:00:00.000 19:00:0	0.001	Input: Who painted the Mona Lisa? Output: Leonardo da Vinci	0.		
GPU Utilization	52.8 %	Input: What's the boiling point of water in Celsius? Output: 100 degrees Celsius	Θ.		
		Input: What's the square root of 64? Output: 8	Θ.		
		Input: What's the largest organ in the human body? Output: Stomach	0.		
	19:00:00:005 19:00:00 006	Input: Who is credited with inventing the telephone? Output: Alexander Graham Bell	0.		
	10000000				

Fine-tuned to your application

AutoEval works well zero-shot, and can also be fine-tuned per application

Each model can be fine-tuned for your application, to get **customized metrics** and measure performance in the context of your task.

- Fine-tune with an API
- Manage your custom evaluators
- Customize metrics specific to your application.

	O Status II Purpose							
S Fine-Tuned Models								
+		ଝି Fine-Tuning Or	nniSphere					
		✓ Data Gen ○ Training	O Serving					
		Will notify you when Fine-Tun	ing is done and					
		the model is ready to be revie	wed.					
You dont have any fine-tuned models yet. Get								
started by fine-tuning a	base model	@ Sneak Peak	U Kill Progress					
© Base Madela								
		Online		Online		Offline		
Online								
Online						Guartika		
Online OmniSphere OmniSphere	and ambadding	VeracityVeil	en authentia	p-Faithful			classification model designed to detect and fla low-quality or potentially harmful Al-generated	
Online OmniSphere OmniSpehre is an advan generation model design	ed embedding ed to create rich,	VeracityVeil A model to differentiate betwee high-quality content and subs	een authentic, tandard or	p-Faithful classification model designe low-quality AI-generated res	ed to detect and flag sponses and out of	classification model der low-quality or potential	signed to detect and flag ly harmful Al-generated	
Online OmniSphere OrmiSpehre is an advan generation model design multidimensional representation and types.	ed embedding ed to create rich, intations of diverse	VeracityVeil A model to differentiate betwn high-quality content and subs malicious Al-generated mater NLP.	een authentic, tandard or ial using SOTA	p-Faithful classification model designe low-quality Al-generated res context distribution content.	ed to detect and flag sponses and out of	classification model der low-quality or potential content across various of	signed to detect and flag ly harmful Al-generated digital platforms.	

Guardrails are just evaluators that run online

Evaluation and guardrails are 2 sides of the same coin

AutoEval models are fast (< 300ms on CPU), and cheap to operate (1/1000th the cost of GPT-4) \rightarrow you can run them **on every response** as a guardrail.

- *Example*: use AutoEval to measure faithfulness, and if a response is deemed to have hallucinated, route to a backup response instead.
- You can also train custom guardrail models for specific safeguard policies using the same base model



Application performance dashboard



Thank you!

Get in touch with me about LastMile AutoEval

AutoEval Signup email: sarmad@lastmileai.dev

any questions?