1. How do you evaluate the best LLM model for your use case?

Selecting the right model is a multi-step engineering process, not just looking at a leaderboard.

- **Step 1: Define Constraints:**
    - **Cost:** Can you afford $30 per 1M tokens (GPT-4) or do you need $0.50 (Llama-3-8B)?
    - **Latency:** Does the user need an answer in 200ms (Voice bot) or 10 seconds (Report generator)?
    - **Privacy:** Can data leave your VPC? (If no, SaaS models are out).
- **Step 2: Create a "Golden Dataset":**
    - Curate 50-100 high-quality examples of (Input, Ideal Output). This is non-negotiable. Without a ground truth, you are just guessing.
- **Step 3: Run the "Vibe Check" (Qualitative):**
    - Manually test 5-10 complex queries on candidates (e.g., Claude 3.5 Sonnet vs. GPT-4o vs. Llama 3).
- **Step 4: Scale Evaluation (Quantitative):**
    - Run the full Golden Dataset through the candidates. Use **LLM-as-a-Judge** (e.g., use GPT-4 to grade the answers of Llama-3) to score them on accuracy, tone, and format compliance.

---

2. How to evaluate RAG-based systems?

RAG systems have two failure points: **Retrieval** (finding the right data) and **Generation** (answering correctly). You must evaluate them separately.

The "RAG Triad" Framework:

1. **Context Relevance (Retrieval):** Did the search engine return *only* useful information, or did it return noise?
2. **Grounding/Faithfulness (Generation):** Is the answer derived *purely* from the retrieved context, or did the model hallucinate/use outside knowledge?
3. **Answer Relevance (Generation):** Did the model actually answer the user's question?

Tools: The industry standard is **Ragas** (Retrieval Augmented Generation Assessment) or **DeepEval**.[1] These frameworks automate the scoring of these three pillars.[2]

+1

---

3. What are different metrics for evaluating LLMs?

Metrics fall into three categories:

A. Statistical / N-Gram Metrics (Legacy)

- **BLEU / ROUGE:** Measures word overlap between the generated text and reference text.
- *Why they fail:* "The cat is on the mat" and "On the mat sits the cat" have low overlap but identical meaning. **Avoid these for chat/reasoning tasks.**

B. Model-Based Metrics (The Modern Standard)

- **LLM-as-a-Judge:** Using a stronger model (e.g., GPT-4) to score the output of a weaker model on a scale of 1-5 based on specific criteria (e.g., "Helpfulness", "Safety").
- **BERTScore:** Uses embeddings to measure semantic similarity, not just word overlap.
- **Perplexity:** Measures how "surprised" a model is by a text. Lower is better. (Mostly used for pre-training/fine-tuning evaluation, not for chatbots).

C. Operational Metrics

- **Tokens Per Second (TPS):** Measures inference speed.[3]
- **Time to First Token (TTFT):** Critical for perceived latency in streaming apps.[4]

---

4. Explain the Chain of Verification (CoVe).

Chain of Verification (CoVe) is a prompting pattern designed to reduce hallucinations.[5] It forces the model to fact-check itself before giving a final answer.

The 4-Step Process:

1. **Draft:** The model generates an initial, baseline response to the user's query.
2. **Plan Verification:** The model generates a list of fact-checking questions based on its own draft.
    - *Draft says:* "The iPhone was released in 2005."
    - *Verification Question:* "When was the first iPhone actually released?"
3. **Execute Verification:** The model answers those specific questions independently (often using RAG or its own knowledge) to see if the draft was correct.
4. **Final Refinement:** The model generates the final verified answer, correcting any mistakes found in step 3.

- **Why it works:** LLMs are essentially autocomplete engines.[6] Once they write a wrong fact, they tend to "double down" on it to stay consistent. CoVe breaks this cycle by forcing a pause and a self-critique loop.[7]
- +1