# DeepScribe Evaluation Suite Write-up

## Goals

At DeepScribe, we need to:

1. **Move fast** - We need to be able to quickly measure and incorporate the latest models and PR changes, without waiting days/weeks.
2. **Understand production quality** - It’s important for us to measure our note quality in the wild, so we can quickly detect any regressions or areas where notes may have lower quality.

## Approach

### 1. Move Fast

To achieve rapid iteration, my evaluation suite was designed with speed, modularity, and reproducibility in mind:

* **Deterministic Metrics**: I implemented lightweight string- and regex-based extractors that compute precision, recall, F1, BLEU/ROUGE, hallucination counts, and contradiction checks. These run within seconds on hundreds of notes, making them suitable for CI/CD pipelines.
* **Synthetic Proxy Data**: The proxy\_model.py tool generates corrupted versions of reference notes at different severities (“mild, medium, spicy”). This allows us to validate our evaluators without waiting for full model inference runs and provides immediate regression testing capability.
* **Flexible LLM-as-a-Judge Backend**: The suite supports multiple backends for semantic evaluation:
  + none → deterministic metrics only (fast, default in CI).
  + openrouter → semantic evaluation via an open LLM API.
  + openai (optional) → supports OpenAI APIs for richer analysis.  
    This modularity enables fast iteration day-to-day, while enabling deeper semantic checks on demand.
* **Single Command Execution**: Developers can run one command to evaluate notes end-to-end, producing both per-case reports and a dashboard.

**Result:** We can measure the quality of a new model version or PR within minutes, not days.

### 2. Understand Production Quality

Production evaluation requires not only reference-based metrics but also methods that work without curated references. Our solution combines both:

* **Reference-Based Evaluation**:
  + **Missing Findings (Recall)** -> checks if the note omitted key transcript facts.
  + **Hallucinations** -> flags unsupported statements not present in the transcript.
  + **Contradictions** -> detects negation errors (e.g., “denies pain” vs. “reports pain”).
  + **Precision, Recall, F1, BLEU/ROUGE** -> provide multiple perspectives on overlap between generated notes and references.
* **Non-Reference-Based Evaluation**:  
  With no curated references in production, we use **LLM-as-a-judge**. The judge evaluates:
  + **Completeness** -> Did the note capture all clinically relevant information from the transcript?
  + **Grounding** -> Are all statements supported by transcript evidence?
  + **Clinical Accuracy** -> Are the statements correct, consistent, and medically safe?
* **Blended Overall Score**:  
  For monitoring, we combine deterministic F1 with normalized LLM scores into a single “overall score.” This top-line number is easy to track and interpret while still grounded in multiple dimensions of quality.
* **Dashboards**:  
  Each run produces:
  + Per-case JSON with detailed flags.
  + Summary JSON with aggregate averages.
  + An HTML dashboard that visualizes both deterministic and LLM scores over time.

## Measuring the Quality of the Evaluator

How do we know if my evaluator itself is working?

1. **Synthetic Stress Tests**: By running on proxy data (mild/medium/spicy), we validate that missing, hallucination, and contradiction errors are consistently flagged.
2. **Clinician Agreement**: On sampled notes, we compare evaluator outputs against clinician judgments. Agreement rates measure evaluator reliability.
3. **Outcome Correlation**: Over time, we check whether higher evaluator scores align with downstream improvements in clinician satisfaction, reduced corrections, or throughput gains.

## Creativity & Extensions

* **Stress-Testing with Proxy Data** ensures the evaluator catches realistic classes of errors.
* **OpenRouter Integration** provides semantic judgment at scale without local GPU or OpenAI quota dependencies.
* **Flexible Backend Switching** future-proofs the system: as stronger open-source LLMs emerge, we can incorporate them easily.
* **Blended Metrics** bridge deterministic rigor with semantic insight, producing a balanced overall signal.

## Conclusion

This dual-track evaluation framework allows DeepScribe to **move fast** with lightweight, reproducible metrics, while also **understanding production quality** through layered reference-based and LLM-based evaluations. By validating the evaluator itself via synthetic stress tests and clinician agreement, I ensure that our metrics remain trustworthy indicators of note quality. This approach empowers DeepScribe to ship quickly while maintaining the clinical trust essential for real-world deployments.