**Typically, the instrumentation for LLM Observability is done in 2 ways:**

1. **Tracing -** Trace the calls you make (with or without an agent) and create spans for request, response data to create metrics and do quality analysis. This is useful for cases where you would want to only look at what happened and don’t want any interference in your usage
2. **Proxy** - Replace the LLM calls to a proxy layer which intercepts your calls and make those calls on your behalf. These help in controlling some trivial parameters without you worrying about them but could lead to application downtime in case the proxy is down.

**Benefits:**

**1. Cost optimisation** for usage:

* The tool would provide you input/output tokens volume for every request being sent to an LLM. This helps you look at the cost overhead of your request, both at an individual level or aggregate.**‍**
* **Custom Tagging:** In case you are looking for attributing cost to different entities, by adding tags for use-cases / accounts.

**2. A/B Experimentation on Prompts / Prompt Engineering:** Try out multiple prompt templates, keep them side by side and explore which prompt give best results and iterate on them. What would otherwise require extensive manual interpretation on a UI layer like chatGPT, will be done programmatically and tracked in the LLM Observability tool.

**3. Evaluate the right LLM for your use-case:** The below screenshot shows an output from [empirical.run](http://empirical.run/) — an open source project shows the comparison between different models and their outputs for given prompts. Such comparisons are generally available within LLM Observability tools and help you accelerate your journey to identifying the best model for your use-case.

**4. Improve RAGs:** An LLM Observability tool will also help you efficiently iterate over your context retrieval based queries, w.r.t prompts, chunking strategy and vector store format.

**5. Fine-tuned model performances:** Analyse how training the LLM with your data is affecting outputs and if there’s any significant different on the output quality between different versions of your model.

**Open-source tools list:**

* Trulens - open source, python only (fully open source no paywalls)
* Phoenix - freemium model, open-source(free if locally hosted)
* Traceloop - open source, free 10 k traces monthly (upto 50 k tokens monthly)
* Langfuse: freemium plans -- <https://langfuse.com/pricing> (upto 50 k tokens monthly)