Google Summer of Code 2025 Proposal

Project Title: Improve Evals Documentation for the Gemini APIs
Contributor: Ria Kabra
Contact : +91 7013826329

Email: riakabra1@gmail.com

1. Synopsis

This project aims to improve the evaluation experience for developers working with Google's Gemini models by enhancing documentation and providing comprehensive code examples. The focus is on clarifying model capabilities, configuration options, and evaluation practices using tools like wandb and promptfoo. This work will bridge the gap between Gemini and open-source LLMs, making Gemini more accessible to researchers and developers.

2. Benefits to the Community

While Gemini is a powerful multimodal LLM, its documentation — especially around evaluation and experimentation — lags behind that of open models like Mistral and Claude. This project addresses key pain points:

- Making Gemini models easier to experiment with
- Providing structured, reproducible examples of prompt evaluations
- Enabling developers to visualize, compare, and debug Gemini responses
- Adding Gemini-friendly config files and documentation for promptfoo and wandb

It contributes toward making the Gemini ecosystem more developer-friendly and research-ready.

3. Deliverables

Phase	Deliverable
Phase 1	docs/model_reference.md – Model types, capabilities
Phase 2	docs/config_guide.md – Effect of temperature/top_p etc.
Phase 3	.env.example, API key setup guide
Phase 4	image_prompt_demo.py, docs for multimodal usage
Phase 5	gemini_vs_mistral_eval.py with wandb logging and plots

Phase	Deliverable
Final	README.md, promptfoo YAML config for Mistral, proposal PDF

4. Technical Approach

Phase 1: Model Format Documentation

Created a clean and complete table listing Gemini models (gemini-2.0-pro, gemini-2.0-flash) with:

- Modalities (text, image)
- Max context size
- Speed vs quality notes

Phase 2: Configuration Options

Demonstrated how changes in:

- temperature
- top_p
- stopSequences

affect output behavior. Used real prompt examples and provided both explanation + code.

Phase 3: Secure Environment Variables

Created a .env.example and explained:

- How to load GOOGLE_API_KEY safely
- Usage in local dev vs cloud

Phase 4: Advanced Features

Used Gemini 1.5 with:

- image = genai.upload_file(path=...)
- Response + image input in the same prompt

Wrote image_prompt_demo.py and documented how to build on it for tool usage, rate limiting, etc.

Phase 5: Evaluation (wandb + promptfoo)

Built:

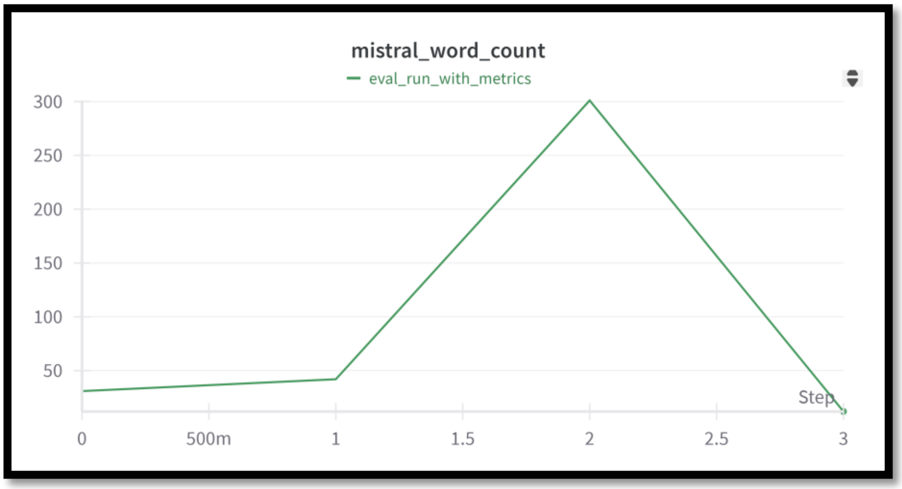
- A prompt list of 4 real-world tasks

  Prompt –

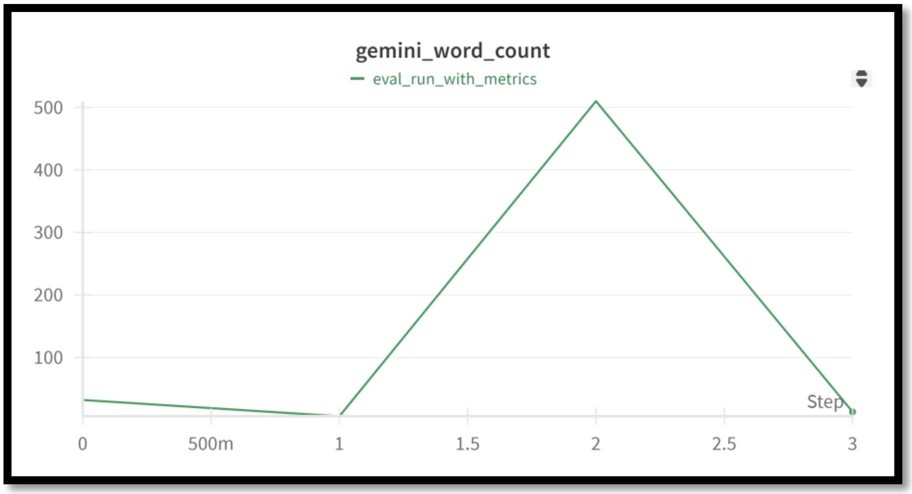
  ```
  prompts = [
      "Summarize the importance of data privacy in one sentence.",
      "What's the capital of Australia?",
      "Explain recursion to a 12-year-old.",
      "Write a haiku about the ocean."
  ]
  ```

- Gemini vs Mistral comparison via ollama.chat()
- Scored responses with:
  - Word count
  - Readability (Flesch score)
- Logged results to wandb

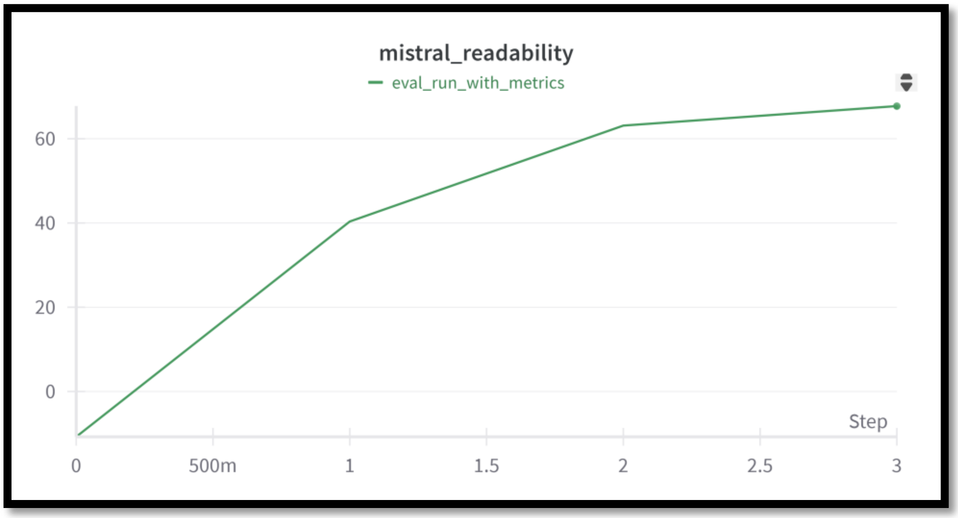
Included screenshots and generated bar/line plots to visualize model differences.



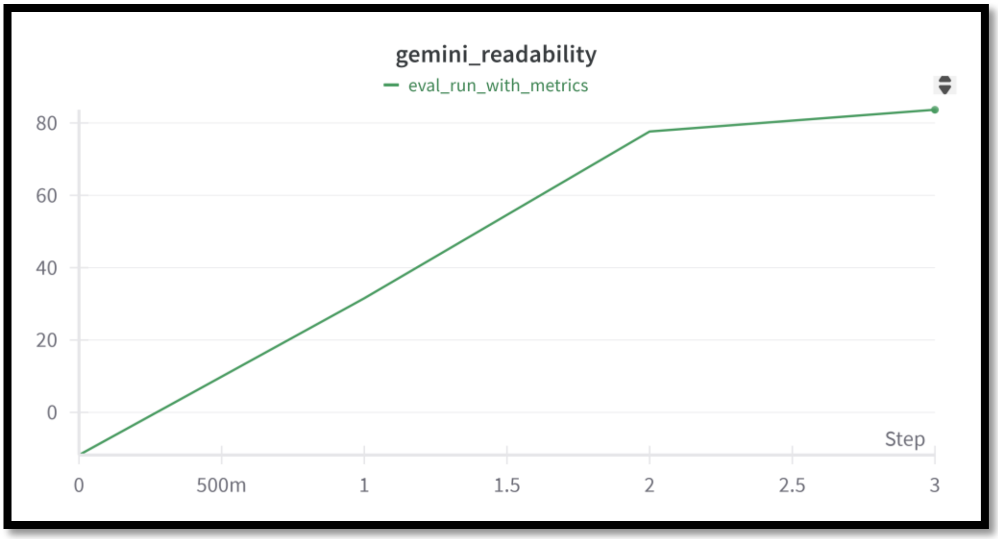
This chart represents the number of words generated by the **Mistral model** across 4 different prompts. The second prompt resulted in a significantly longer response (~300 words), while other prompts were much shorter. This shows that Mistral can be verbose depending on the prompt structure, but its outputs are generally concise compared to Gemini.



This chart shows the **Gemini model's** word count on the same set of prompts. Notably, the second response exceeds 500 words, indicating Gemini's tendency to provide detailed answers. The overall word count is higher than Mistral across the board, highlighting that Gemini prefers completeness and elaboration.



This graph tracks the readability of Mistral's responses using the **Flesch Reading Ease Score**. The readability gradually increases from 40 to over 65, which means later responses were simpler and easier to understand. Scores in the 60–70 range are considered "plain English," suitable for a wide audience.



Gemini's readability scores consistently outperform Mistral's. The model achieved scores up to ~85, which indicates very high clarity — readable by middle school students. This aligns with Gemini's strength in structuring outputs in a human-friendly and explainable way, especially useful in education, helpbots, and summaries.

For promptfoo:

- Created a working promptfoo.yaml with Mistral
- Noted that Gemini is not natively supported yet, but documented how it could be

---

5. Timeline

Period	Deliverables
Community Bonding (May)	Read Gemini + promptfoo docs, refine plan
Week 1–2	Phase 1 & 2: Model reference + config tests
Week 3–4	Phase 3 & 4: Environment setup, multimodal examples
Week 5–6	Phase 5: Evaluation + wandb logging
Final Weeks	README, final PDF writeup, polish, blog post

---

6. Why Me

I'm a beginner in open source, I learn by building. I've already built a working version of this project using:

- Gemini's Python SDK
- Ollama's local Mistral model

- wandb for evaluation tracking

I’m excited about LLMs and want to help make Gemini easier for others to adopt.

---

7. Links

- □ GitHub Repository: https://github.com/RiaKabra/gemini-evals-docs/
