DRAWING WITH LLMS

AUTHORS

Jose Fernando Ramirez Ortiz: 20241020080
Carlos Andres Brito Guerrero – 20241020147
Daniel Alonso Chavarro Chapatecua – 20241020066

INTRODUCTION

Kaggle's "Drawing with LLMs" competition challenges participants to generate SVG (Scalable Vector Graphics) images from brief text descriptions of up to 200 characters. The generated SVGs must comply with strict constraints: a maximum size of 10,000 bytes, no CSS styling, no external fonts, and no rasterized image data. The primary challenges lie in ensuring compliance with these technical limitations while achieving high visual fidelity and mitigating the inherent variability of generative language models.

OBJECTIVES

- Generate valid SVG images from short textual descriptions using LLMs.
- Comply with strict format constraints, including size limits and disallowed elements (CSS, raster images, etc.).
- Improve visual quality and consistency through prompt engineering and template variation.
- Evaluate system performance across multiple categories and detect common generation errors.

PROBLEM STATEMENT

How can we design and implement an efficient, reusable system that uses Large Language Models (LLMs) to generate high-quality, valid SVG images from short textual descriptions, under strict format and size constraints, and without relying on internet access or proprietary models?

METHODOLOGY

1. System Architecture Design: Developed a modular system with key components: Input Handler, Prompt Generator, LLM SVG Generator, and Evaluation Module.
2. Prompt Engineering: Designed and tested three template versions to optimize SVG generation quality and compliance.
3. API Integration: Integrated Google Gemini 2.5 Flash via API for SVG code generation, with safety filters and constraint checks.
4. Simulation and Testing: Used 60 categorized prompts (landscape, fashion, abstract, unknown) to test performance across templates and identify constraint violations.
5. Output Evaluation: Measured compliance rate, file size, and generation errors (e.g., parse issues, invalid elements) to assess effectiveness.

RESULTS

Performance Overview

- Total Prompts Tested: 60
- Prompt Templates Evaluated: 3 (v1, v2, v3)
- Categories Tested: Abstract, Fashion, Landscape, Unknown
- Overall Compliance Rate: 76.7%
- Average SVG Size: ~1,992 bytes
- Average Generation Time: 135 ms

ANALYSIS

TEMPLATE PERFORMANCE:

Template	Compliance Rate	Avg. File Size (bytes)	Error Rate
v1	70%	1,995	30%
v2	**80%**	**1,893**	**20%**
v3	80%	2,087	25%

INSIGHT:
TEMPLATE V2 SHOWED THE BEST TRADE-OFF BETWEEN OUTPUT QUALITY AND FILE SIZE, MAKING IT THE RECOMMENDED VERSION FOR DEPLOYMENT.

CATEGORY-SPECIFIC PERFORMANCE:

Category	Compliance Rate	Error Rate
Landscape	**85.70%**	14.30%
Fashion	83.30%	16.70%
Abstract	66.70%	**38.10%**
Unknown	66.70%	33.30%

INSIGHT:

- ABSTRACT AND UNKNOWN CATEGORIES PERFORMED WORSE DUE TO VAGUE DESCRIPTIONS.
- LANDSCAPE AND FASHION CATEGORIES YIELDED MORE CONSISTENT SVGS, LIKELY DUE TO THEIR STRUCTURED VISUAL NATURE.

CONCLUSION

We successfully designed a modular, cost-effective system capable of generating SVGs from text using LLMs. While full implementation is ongoing, initial testing shows the feasibility of our approach. Prompt engineering proved critical for quality output. The system answers the research question by balancing performance, fidelity, and constraint handling.