

API Documentation Optimizer using TextGrad

DA312 Course Project

Himanshu Singhal

220150004

3rd Year BTech, DSAI

IIT Guwahat

Abstract—This project aims to adapt TextGrad’s ”automatic differentiation via text” framework to create an automated API documentation optimizer using open-source Large Language Models (LLMs). We implement TextGrad’s iterative feedback methodology - evaluate, generate feedback, and improve - specifically for technical documentation enhancement using models like Llama 3.1. The system focuses on improving documentation completeness, technical accuracy, and usability across a test dataset of real-world API documentation from open-source projects. I hope to evaluate this on publicly available API documentation such as OpenAPI and KaggleAPI. Evaluation is intended to be done mostly using LLM-as-a-Judge with well-defined metrics in LangSmith. This work aims to demonstrate practical application of TextGrad with open-source LLMs while contributing insights into text-based optimization for technical documentation.

I. INTRODUCTION

TextGrad has emerged as a powerful framework for optimizing AI systems through natural language feedback, yet its application to technical documentation remains unexplored. API documentation, crucial for developer experience and software maintainability, often suffers from inconsistency and incompleteness. By adapting TextGrad’s methodology to API documentation optimization using open-source LLMs, we aim to address this gap while making the technology more accessible. Our project explores whether automated, iterative improvement of technical documentation can achieve standards comparable to human-written documentation.

II. TEXTGRAD FRAMEWORK

TextGrad’s core concept of using LLMs to generate textual feedback for optimization will be adapted specifically for API documentation. The process involves:

- Evaluating current documentation quality using predefined metrics
- Generating specific improvement suggestions through LLM feedback
- Applying improvements based on this feedback
- Iterating until quality thresholds are met

Using Llama 3.1 as our primary model, I’ll focus on optimizing key documentation components: method descriptions, parameter definitions, return values, and usage examples. The system will maintain TextGrad’s optimization loop while incorporating domain-specific constraints for technical accuracy.

III. DATASETS

The project will utilize two main sources of API documentation:

- **OpenAPI Specifications:** Public API documentation from major platforms, providing structured documentation in a standardized format
- **Kaggle API:** Documentation from a widely-used data science platform, offering real-world examples of technical API documentation

These datasets represent different documentation styles and complexity levels, allowing us to test our system’s versatility. The documentation will be preprocessed to extract key components for optimization while maintaining structural integrity.

IV. EVALUATION METHODOLOGY

Our evaluation approach centers on LLM-as-a-Judge using LangSmith, focusing on three key aspects:

- **Documentation Completeness:** Coverage of essential elements (parameters, returns, examples)
- **Technical Accuracy:** Correctness of type information and parameter descriptions
- **Usability:** Clarity and practicality of examples and explanations

LangSmith will provide structured evaluation pipelines with well-defined metrics, enabling consistent assessment of improvements across different documentation samples.

V. CONCLUSION

This project aims to demonstrate the practical application of TextGrad principles in technical documentation optimization using open-source LLMs. By focusing on API documentation and implementing a robust evaluation framework, we hope to contribute both practical tools for documentation improvement and insights into applying text-based optimization techniques in technical contexts. Success in this project could pave the way for broader applications of TextGrad in software development workflows.