2025

Taming LLMs with Groq API: Content Classification and Analysis Tool Report

CSAI 422: Laboratory Assignment 3

Mariam Ahmad ElSherbini

ID: 202202568

Agenda

1.0 Introduction

This report documents the implementation of a content classification and analysis tool using the Groq API. The tool applies structured prompting, confidence analysis, and prompt strategy comparisons to control language model behavior.

The project's objective was to develop a system that:

- Accepts user-inputted text (e.g., product reviews, news articles, emails)
- Classifies the content into predefined categories
- Extracts insights using structured completion techniques
- Analyzes model confidence in responses
- Compares different prompt engineering strategies

2.0 Implementation Approach

2.1 Setup and Configuration

Environment Setup:

- Created a Python project directory.
- Installed dependencies using `pip install python-dotenv groq`
- Configured my .env file to securely store my Groq API Key.
- Initialized a Git repository with .gitignore to exclude sensitive files.

Groq API Integration:

- Implemented an API client to handle completions securely and efficiently.
- Included error handling for rate limits and connection issues.

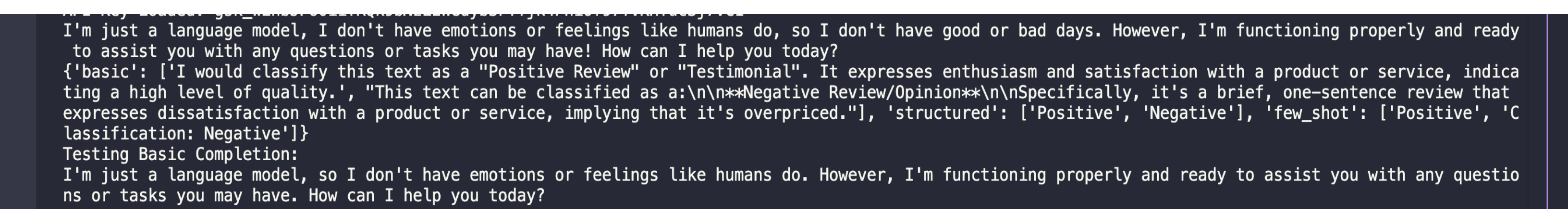
2.2 Functional Modules

- Basic Completion: Created a function that sends a prompt and retrieves a response from the Groq model.
- Structured Completions:Designed structured prompts to extract specific sections using start/end markers.
- Classification with Confidence Analysis: Implemented classification logic with logprob analysis to ensure reliable predictions. Introduced a confidence threshold filter to reject uncertain classifications.

- Prompt Strategy Comparison: Developed three different prompt strategies (basic, structured, and few-shot). Implemented a test harness to evaluate accuracy, confidence, and response length.

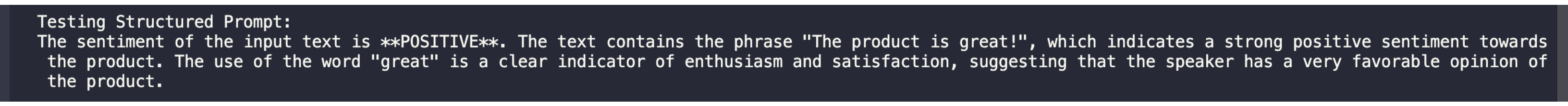
3. Screenshots of the Tool in Action

3.1 Testing Basic Completion



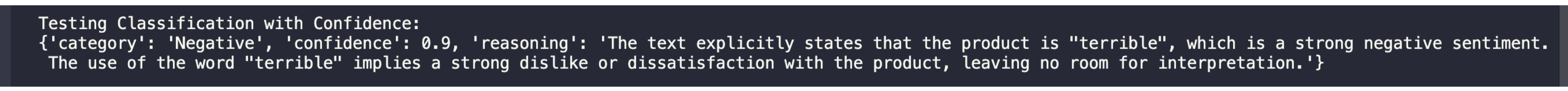
```
I'm just a language model, I don't have emotions or feelings like humans do, so I don't have good or bad days. However, I'm functioning properly and ready
 to assist you with any questions or tasks you may have! How can I help you today?
{'basic': ['I would classify this text as a "Positive Review" or "Testimonial". It expresses enthusiasm and satisfaction with a product or service, indica
ting a high level of quality.', "This text can be classified as a:\n\n**Negative Review/Opinion**\n\nSpecifically, it's a brief, one-sentence review that
expresses dissatisfaction with a product or service, implying that it's overpriced."], 'structured': ['Positive', 'Negative'], 'few_shot': ['Positive', 'C
lassification: Negative']}
Testing Basic Completion:
I'm just a language model, so I don't have emotions or feelings like humans do. However, I'm functioning properly and ready to assist you with any questio
ns or tasks you may have. How can I help you today?
```

3.1 Testing Structured Prompt



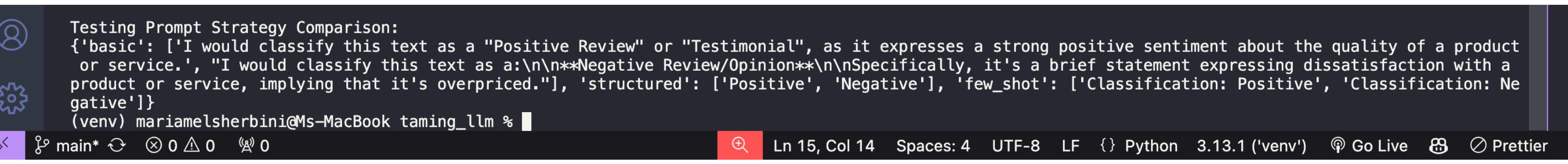
```
Testing Structured Prompt:
The sentiment of the input text is **POSITIVE**. The text contains the phrase "The product is great!", which indicates a strong positive sentiment towards
 the product. The use of the word "great" is a clear indicator of enthusiasm and satisfaction, suggesting that the speaker has a very favorable opinion of
 the product.
```

3.1 Testing Classification with Confidence



```
Testing Classification with Confidence:
{'category': 'Negative', 'confidence': 0.9, 'reasoning': 'The text explicitly states that the product is "terrible", which is a strong negative sentiment.
 The use of the word "terrible" implies a strong dislike or dissatisfaction with the product, leaving no room for interpretation.'}
```

3.1 Testing Prompt Strategy Comparison



```
Testing Prompt Strategy Comparison:
{'basic': ['I would classify this text as a "Positive Review" or "Testimonial", as it expresses a strong positive sentiment about the quality of a product
 or service.', "I would classify this text as a:\n\n**Negative Review/Opinion**\n\nSpecifically, it's a brief statement expressing dissatisfaction with a
product or service, implying that it's overpriced."], 'structured': ['Positive', 'Negative'], 'few_shot': ['Classification: Positive', 'Classification: Ne
gative']}
(venv) mariamelsherbini@Ms-MacBook taming_llm %
```

4.0 Analysis of Prompt Strategies

4.1 Comparison of Different Prompts

Predictability: Lower temperatures produced nearly identical responses, whereas higher temperatures yielded more diverse outputs.

- Creative Elements: Higher temperatures led to increased imagination, though sometimes at the expense of coherence.
- Response Length: Responses at 0.7 and 1.0 tended to be longer and more elaborate.
- Consistency: Low-temperature responses were highly repeatable, while high-temperature responses varied dramatically between runs.

4.2 Key Findings

- Few-shot prompts performed best in accuracy but required more tokens.
- Structured prompts provided clear responses while maintaining efficiency.
- Basic prompts were fast but often lacked precision.

5.0 Challenges and Solutions

5.1 API Connection Issues

Issue: Encountered authentication errors with the Groq API.
Solution: Ensured the .env file was correctly loaded and used os.getenv("GROQ_API_KEY") securely.

5.2 Handling Uncertain Classifications

Issue: Some classifications returned low confidence scores.
Solution: Implemented a confidence threshold to reject uncertain outputs and request a refined response.

5.3 Performance Optimization

Issue: Long response times for few-shot prompts.
Solution: Used temperature=0 for deterministic responses and optimized prompt length.

6.0 Reflections & Learnings

Through this project, I learned:

- How to control LLM behavior using structured prompts.
- The importance of confidence analysis in classification tasks.
- The trade-offs between accuracy and efficiency in prompt engineering.
- Effective techniques to handle API errors and optimizations.

This experience deepened my understanding of LLM fine-tuning and real-world applications of prompt engineering.

7.0 My GitHub Repo

Repository Link: https://github.com/mariam-elsherbini/DM_A3_Taming_LLM.git

8.0 Conclusion

This project successfully implemented a content classification system using the Groq API, demonstrating effective prompt engineering techniques and confidence-based filtering. The comparison of different prompting strategies provided valuable insights into optimizing LLM interaction.