Ali Usama

Prof. Eduardo

Deep Learning ECBS5200

24 May 2025

Prompt-Engineering for Customer-Review Analysis

Methodology

To start with I have used the IMDb Pulp Fiction reviews dataset. For all three tasks, sentiment classification, key praise or complaint extraction, and two-sentence summarization, I developed three successive versions of prompts, iteratively refining with a 20-review validation subset. Version 1 established a plain English baseline. Version 2 added explicit rules like negative keywords, single-sentence limits and length caps. Version 3 added a few-shot exemplars when rule-based tuning plateaued. Both GPT-4.1-nano by OpenAI and Gemini Flash by Google were evaluated on each version for each task. Following each round, I created a manual label column and then computed automatic metrics like Macro-F1 on sentiment, token-level F1 on phrase extraction and ROUGE-L for summaries. Furthermore, I have manually explored failures before iteratively adjusting the subsequent prompt.

Findings and Best-Performing Combinations

Task	Best Combo	Score
Sentiment	GPT-4.1-nano + v2	Macro-F1 0.589
Key Praise / Complaint	GPT-4.1-nano + v3	Token-F1 0.503
Summary	Gemini Flash + v2	ROUGE-L 0.216

For sentiment task, GPT-4.1-nano with sentiment_v2 did best as Macro-F1 was 0.59. The prompt's short list of negative cue words ("worst", "skip it"), a rule for even praise and criticism, and sarcasm instructions corrected the positive-bias seen in v1 without suffering from the Neutral over-priming that the few-shot examples in v3 did. Moreover, for key praise or compliant extraction, GPT-4.1-nano with snippet_v3 performed best as token-F1 was 0.50. A rule of three diverse examples and a "select the one sentence with the most emotional language" rule helped the model to disregard boring starts and focus on the most descriptive complaint or compliment. Finally for summaries, Gemini Flash with summary_v2 performed best among the pairs as ROUGE-L was 0.216. The strict two-sentence, ≤ 35-word directive, coupled with a paraphrasing requirement, resulted in short summaries that were on topic and did not copy. GPT-4.1-nano's v3 prompt, while close in ROUGE, incurred more length violations. Leaderboards for all versions are included in the notebook.

Failure Patterns and Root Causes

Across tasks the models struggled most with ambiguous or edge-class inputs. For sentiment, neutral reviews that contained an equal balance of praise and criticism were consistently mislabeled. A single rule in v2 helped but didn't eliminate the bias, suggesting a need for more domain-specific examples or class-balanced fine-tune. In key praise/complaint extraction, truncation occurred occasionally when the emotional sentence exceeded the generation limit. Raising the limit and filtering on terminal punctuation reduced the error but did not completely remove it. Summaries occasionally resorted to formulaic language ("A glowing review praising…"), a consequence of few-shot priming and the strict compression requirement.

Metric Limitations

Macro-F1 and ROUGE with only twenty manually labelled examples are error-prone; a single mis-prediction alters scores by a few percentage points. Token-level F1, while less sensitive than exact match, but over-values greatly short snippets whose words happen to overlap. ROUGE-L calculates n-gram overlap and does not detect factual accuracy; a factually inaccurate but fluent summary will still score mid-range. To remedy these short-coming, I manually checked mismatches and on summaries recorded anecdotal fidelity/fluency notes.

Cost, Latency, and Deployment Confidence

Gemini Flash (free plan, ~0.2 s/call) suited the task of summarizing by volume perfectly, but GPT-4.1-nano (~0.8 s/call, ~$0.002/1k tokens) gave stronger, more predictable structured extraction of praise/complaint at a decent price. The throttled run, always trying Gemini first, and only spending Open AI tokens on completed prompts, saved total spend by about 60 %. In production I would publish the winning pipelines with grade B confidence: good enough for dashboard aggregation and user-facing snippets but still requiring human review of neutral sentiment cases and periodic spot-checking of summary accuracy.

Lessons Learned

During this project, not only did I become proficient in prompt engineering techniques but also in programmatically integrating and processing two different foundation-model APIs (Open AI's GPT-4.1-nano and Google's Gemini Flash) programmatically, including applying a throttling mechanism to respect Gemini's free-tier rate limits. One of the key observations was that minimalism is rewarded early on: short, rule-based prompts (such as our "v2" variants) would outperform longer few-shot prompts when class distribution was skewed, as they focus the model on the critical decision factors without over-priming. At the same time, few-shot is powerful but perilous. Adding too many exemplars or reinforcing one class too much may induce template bias or degrade essential rules, so exemplars must be used carefully. I also came to understand the importance of metric choice: precision on exact-match accuracy painted a somber portrait (<0.2) for snippet extraction, yet token-level F1 experienced more nuanced partial matches (>0.5), calling for supplementing automated metrics with human assessment. To manage costs and quotas, I used a quota-aware workflow, making Gemini calls first under customized

throttle and calling Open AI only after prompts were finished. This cut approximately 60 % of total token expenditure. Finally, I learned that one prompt doesn't fit all tasks: sentiment classification and key praise/complaint extraction thrived on straightforward rule lists, while summarization demanded strict sentence and word-count caps plus paraphrase constraints to balance brevity and fidelity.

Conclusion

With iterative prompt engineering, careful selection of metrics, and quota-aware evaluation, I demonstrated that foundation models can deliver actionable customer feedback insights with minimal code. Follow-up work will optimize a small open-weights model for neutral recall sentiment, add human fluency scores to summary evaluation, and incorporate lightweight post-generation heuristics to defend against truncation and template vocabulary.