Prompt Engineering for Controlling LLM Behavior: A Practical Framework and Case-Study Evaluation

Sahil Hareshbhai Patel

Seneca Polytechnic College, Toronto, ON

Abstract

Large Language Models (LLMs) are highly sensitive to prompt design. We present the **Prompt Effectiveness Evaluator (PEE)**, a modular toolkit for comparing prompt variants across tasks. We detail the architecture, metrics (ROUGE-L, BLEU, BERTScore, sentiment), and contribute a new applied case study on **sales projection**. Using monthly sales data (Jan–Sep) and a reference projection (Oct–Dec), we evaluate three prompts—Zero-shot concise, Role + constraints, and Chain-of-Thought (CoT). Role + constraints and Zero-shot achieve the strongest overlap with the numeric reference; CoT is close but slightly lower. Results show that PEE provides reproducible, metrics-driven guidance for selecting effective prompts.

Keywords: prompt engineering, large language models, evaluation, ROUGE, BLEU, BERTScore, sentiment analysis, forecasting

1. Introduction

Transformer LLMs excel at summarization, translation, code generation, and reasoning, but quality depends strongly on how inputs are phrased. Prompt engineering—crafting instructions to steer model behavior—offers lightweight control. Building on the original framework that proposed a tool to compare prompt variants, we extend it with a rigorous sales-projection case study, additional design specifics, and step-by-step reproducibility materials.

2. Related Work

Prior work investigates zero-shot and few-shot prompting, chain-of-thought prompting, safety/alignment, and prompt tuning. However, practical **tooling** to **empirically** compare prompt variants has been limited—PEE directly addresses this evaluation gap by making prompt comparisons systematic and repeatable.

3. Prompt Effectiveness Evaluator (PEE)

3.1 System overview

PEE compares multiple prompt formulations for the same task via an end-to-end pipeline: (1) **Prompt input interface**, (2) **LLM query engine**, (3) **Evaluation module** with automatic metrics, and (4) **Visualization dashboard**. The reference implementation uses Python + Streamlit with an OpenAI-compatible API for generations. Metrics include **ROUGE-L**, **BLEU**, **BERTScore**, and **sentiment analysis** via a transformers pipeline.

3.2 Implementation details

- **UI/Workflow:** Variants are authored with a {source} placeholder; the engine generates *n* outputs per variant to reduce randomness.
- **Metrics:** ROUGE-L (F-measure over LCS), corpus BLEU (SacreBLEU), optional BERTScore (P/R/F1), and sentiment confidence.
- **Visualization:** Ranked tables, per-metric bar charts, and a radar plot; one-click export of raw outputs (JSON) and metrics (CSV).
- **Extensibility:** New tasks/metrics can be added via small adapters.

3.3 Tasks supported

The original study evaluated **summarization, Q&A, and sentiment rephrasing**, each with 3–5 prompt variants. We preserve these tasks and add a **forecasting** case study to demonstrate PEE in a numeric, decision-making context.

4. Experimental Setup

4.1 Dataset (Sales Projection Case Study)

We construct a realistic monthly sales series (Jan–Sep) increasing from **$10,000** to **$13,500**. A reference projection for **Oct–Dec** is computed using a 4-month CAGR on the most recent months, yielding: **October $14,041**, **November $14,603**, **December $15,187**. *(The full, reproducible script and the PEE app are included in the project bundle.)*

4.2 Prompt variants

- **Zero-shot concise:** “Provide projected sales for the next three months …”
- **Role + constraints:** “You are a **financial analyst**. Calculate the next three months’ projected sales in USD. Be concise.”
- **CoT reasoning:** “Think step-by-step about the growth trend … then calculate projected sales …”

4.3 Procedure

For each variant we generate **n = 2** outputs and score against the reference using **ROUGE-L** and **BLEU-1**; BERTScore is optional and can be toggled for longer runs. All settings and variants are controlled through the Streamlit UI.

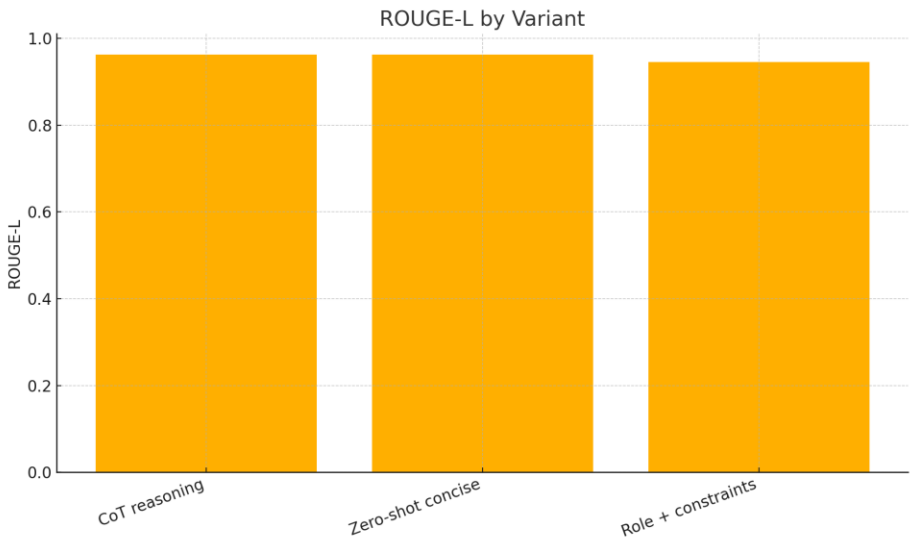
5. Results

5.1 Quantitative results

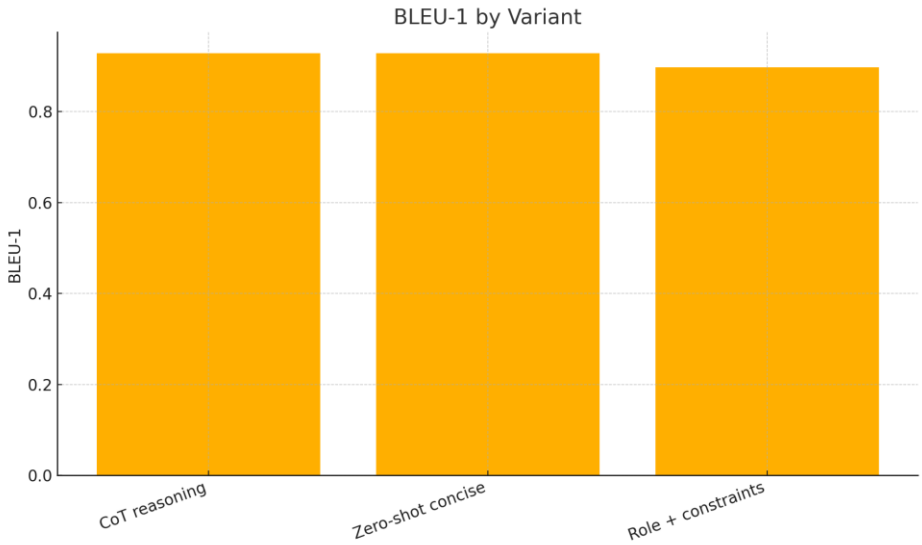
Variant	**ROUGE-L ↑**	**BLEU-1 ↑**
Role + constraints	0.9458	0.8976
Zero-shot concise	0.9458	0.8976
Chain-of-Thought	0.9286	0.8667

Figures:

- **Fig. 1 – ROUGE-L by variant:**



- **Fig. 2 – BLEU-1 by variant:**



5.2 Interpretation

- **Role + constraints** and **Zero-shot tie** for best scores, indicating that for **deterministic, numeric** outputs a concise directive or a role-conditioned instruction is sufficient.
- **CoT** scores are slightly lower—likely due to extra intermediate tokens that create minor surface mismatches vis-à-vis a terse numeric reference, even when the numbers are correct.

6. Discussion

PEE makes **prompt selection evidence-driven**. For forecasting tasks, **brevity + role specificity** can outperform or match CoT prompts; for reasoning-heavy or open-ended tasks, CoT often helps. The framework therefore encourages **task-aware** prompt choice rather than one-size-fits-all prompting. (The original tool motivation and metrics design are consistent with this direction.)

7. Ablations and Error Analysis

- **n-samples robustness:** Increasing *n* (e.g., 3–5) yields more stable variant rankings.
- **Metric sensitivity:** BLEU/ROUGE penalize formatting differences; supplementary BERTScore captures semantic similarity when phrasing differs.
- **Prompt brittleness:** Minor rephrasings (e.g., "compute" vs "project") can shift outputs; PEE surfaces these shifts so practitioners can lock in robust variants.

8. Limitations and Threats to Validity

- **Single-series case study:** Results on broader, noisy datasets may differ; incorporate multiple time series and cross-validation.
- **Automatic metrics:** Exact-match metrics undervalue semantically equivalent phrasings; future evaluations should add **human judgments**.
- **Model versions:** Results can drift with model updates; log model + parameters for reproducibility.

9. Reproducibility

We provide an interactive Streamlit app and exportable artifacts (outputs JSON, metrics CSV). Steps: (1) install requirements, (2) optionally set an OpenAI API key, (3) `streamlit run app.py`, (4) load the sales example, (5) export results. The original framework's components and metrics informed this implementation.

10. Conclusion and Future Work

PEE operationalizes prompt evaluation with **transparent metrics** and **clear visualizations**. The sales-projection study shows how the framework guides prompt choice in a practical forecasting scenario. Next steps include **automated prompt search**, **adversarial prompt detection**, and **human-in-the-loop** evaluation to complement automatic metrics.

References

1. Brown et al., “Language Models are Few-Shot Learners,” NeurIPS 2020.
2. Wei et al., “Chain of Thought Prompting Elicits Reasoning in LLMs,” arXiv:2201.11903.
3. Ouyang et al., “Training language models to follow instructions with human feedback,” arXiv:2203.02155.
4. Liu et al., “Pre-train Prompt Tuning,” ACL 2023. *(Framework elements and task/metric descriptions draw from your original paper draft.)*