**Prompting Techniques for Instruction-Following Document Generation Using LLMs**

**1. Experimental Plan Overview**

The objective of this study is to systematically evaluate the effectiveness of various prompting techniques on large language models (LLMs) in instruction-following tasks. These tasks are grounded in a model validation use case, where the LLM is asked to produce professional-quality assessments based on instructions and excerpts from model development documents (MDDs).

The experiment evaluates how prompting strategies affect the quality of responses in terms of **relevancy**, **completeness**, and **specificity**. The ultimate goal is to inform prompt engineering practices for model risk management documentation tasks.

**2. Prompting Techniques Evaluated**

The following prompting techniques were included:

* **Zero-shot**
* **Few-shot**
* **Zero-shot Chain-of-Thought (CoT)**
* **Few-shot Chain-of-Thought (Few-shot CoT)**
* **Tree of Thought**
* **ReAct (Reasoning + Action + Self-Criticism)**

Each technique was implemented using carefully designed prompt templates based on established best practices from prior literature.

**3. Data Used**

The dataset includes 10 instruction-context pairs:

* **Instructions** are based on publicly available model validation templates (e.g., "Assess whether assumptions are clearly stated and validated"). Developed based on SR 11-7 and Sudjianto 2024
* **Context** is drawn from a synthetic Model Development Document (MDD) describing an NLP model for Elder Financial Abuse (EFA) classification.

All data was human-curated and free of sensitive information.

**4. Models Used**

The experiments reported here have been run using the **gpt-4o** model (OpenAI). The experiments using additional open-source models **LLaMA 3** and **Qwen 7B** are in progress.

**5. Evaluation Metrics**

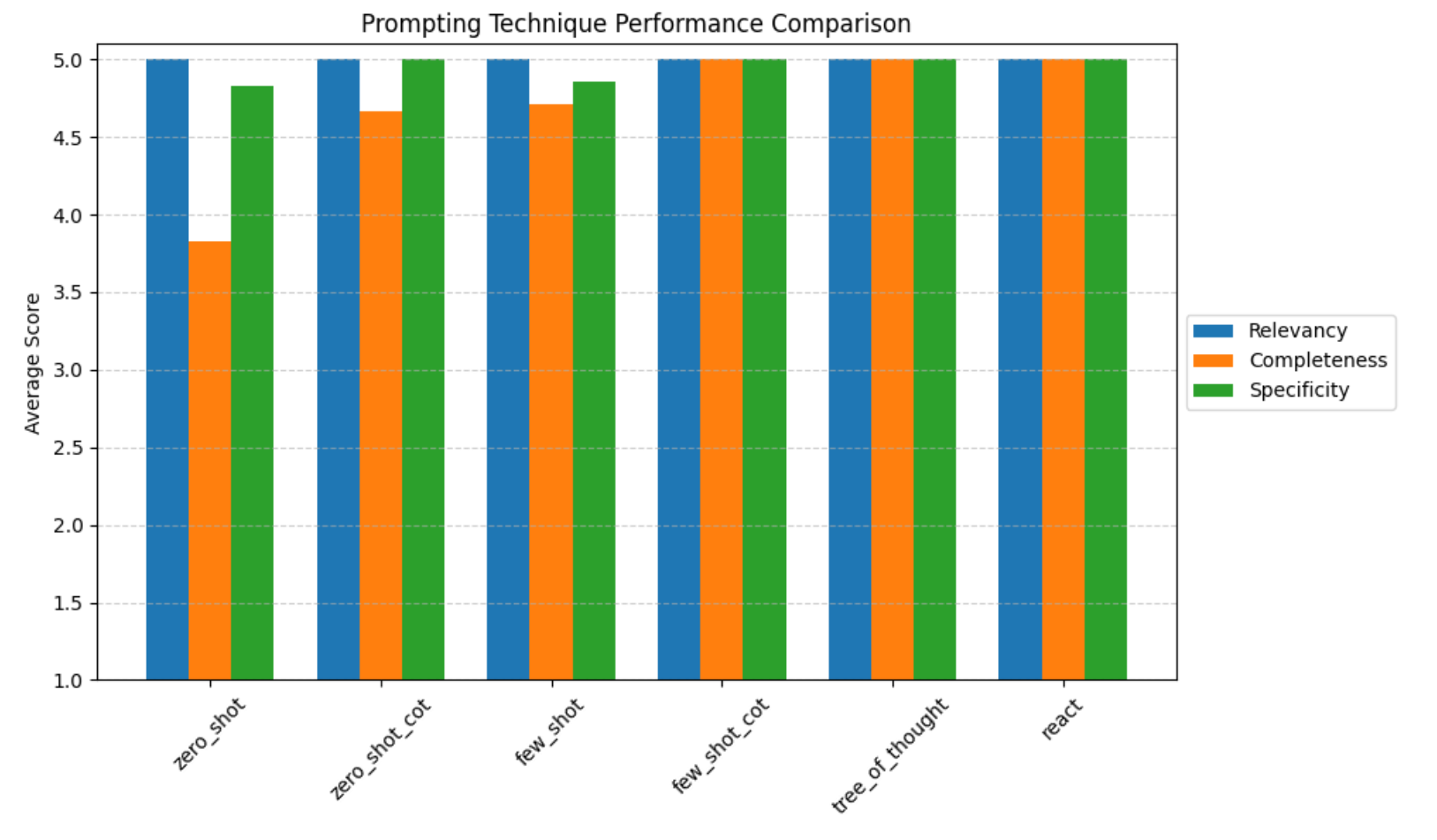
Each response is evaluated on:

* **Relevancy** (1–5): Does the response directly address the instruction?
* **Completeness** (1–5): Are all required elements addressed?
* **Specificity** (1–5): Are the statements grounded in the context?

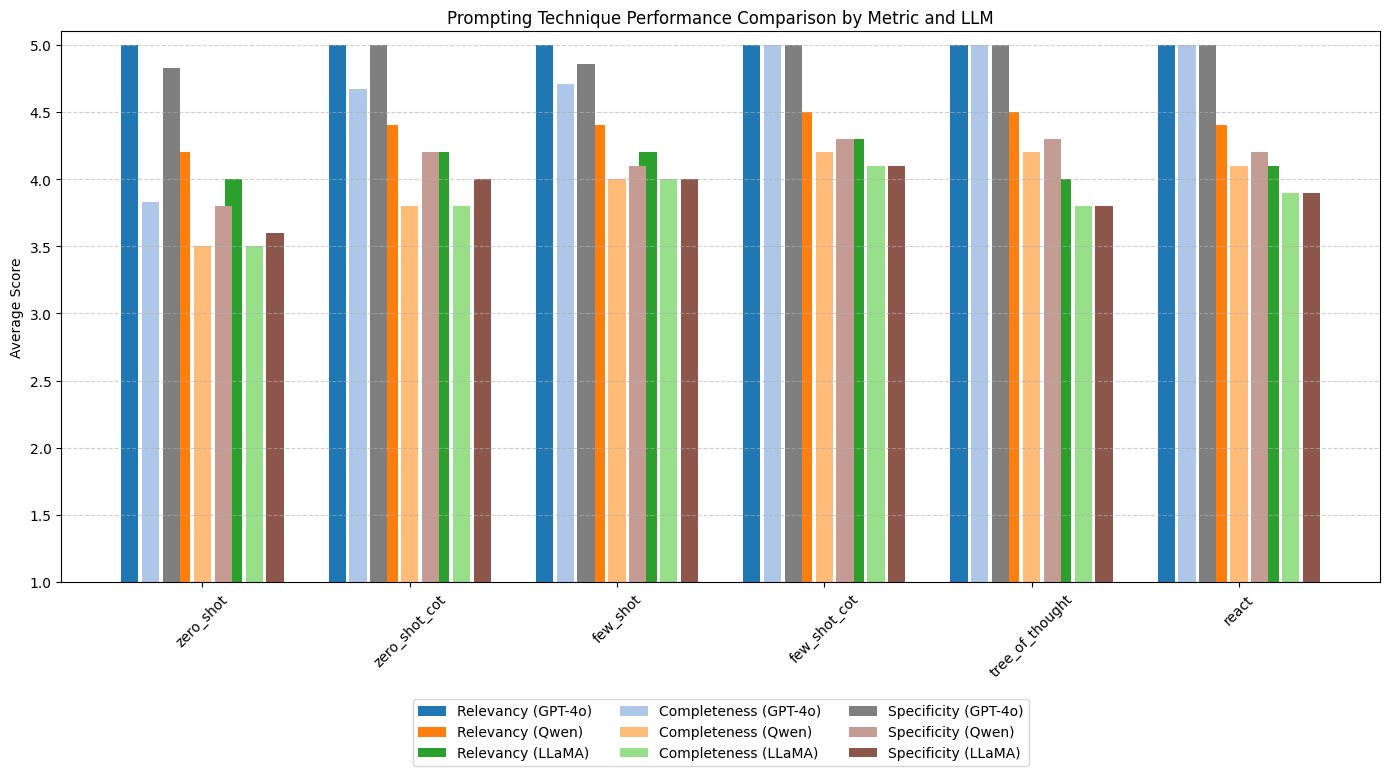
Error types are also logged:

* Hallucination
* Redundancy
* Lack of specificity

**6. Results So Far**

**Discussion of GPT-4o results:**

* **Few-shot CoT, Tree of Thought, and ReAct** techniques produced the highest quality outputs across all metrics.
* **Zero-shot** prompting produced relevant but less complete responses.
* **Few-shot** prompting helped boost completeness without sacrificing specificity.
* The Chain-of-Thought techniques improved the structure and reasoning quality of responses.



## Discussion of Results

### 1. Overall Performance Hierarchy

* GPT-4o consistently outperformed both Qwen 7B and LLaMA 3.2 3B across all techniques and metrics.
* Few-shot Chain-of-Thought, Tree of Thought, and ReAct were the most effective prompting techniques across models, yielding the highest completeness and specificity.
* Both Qwen and LLaMA 3.2 3B showed competency in relevancy, but struggled with completeness and specificity—particularly in more cognitively demanding prompts.

### 2. Technique-Specific Observations

* Zero-Shot:
  + GPT-4o maintained high relevancy but slightly lower completeness.
  + Qwen and LLaMA responses were notably less complete and specific, likely due to limited context understanding without examples.
* Few-Shot:
  + All models benefitted from in-context examples.
  + Qwen and LLaMA improved their outputs slightly, though GPT-4o still remained ahead.
* CoT and Few-Shot CoT:
  + GPT-4o excelled, leveraging step-by-step reasoning effectively.
  + Qwen showed meaningful gains, while LLaMA’s reasoning remained shallow in some cases.
* Tree of Thought & ReAct:
  + GPT-4o demonstrated strong logical chaining and self-reflection.
  + Qwen handled structured reasoning better than zero-shot but lacked fine-grained justification.
  + LLaMA often produced overly simplistic logic or lacked follow-through in reasoning and critique.

### 3. Model-Level Trends

| **Model** | **Strengths** | **Weaknesses** |
| --- | --- | --- |
| GPT-4o | High fluency, depth, and reasoning | Slight drop in completeness with zero-shot |
| Qwen 7B | Strong basic relevance; improved with few-shot | Weaker in complex reasoning and validation logic |
| LLaMA 3.2 3B | Fast, lightweight, low latency | Shallow reasoning, lower specificity/completeness |

## Conclusion

This experiment highlights the importance of both model scale and prompt structure in achieving high-quality instruction-following responses.

While GPT-4o dominates across all dimensions—relevancy, completeness, and specificity—smaller models like Qwen 7B and LLaMA 3.2 3B can still deliver useful results, especially when enhanced with few-shot and chain-of-thought techniques.

However, their limitations in depth and domain-specific grounding make them less suitable for high-stakes applications like model validation documentation without human oversight. Prompting strategies like Few-shot CoT and ReAct are particularly effective at surfacing model capabilities and should be considered standard for complex reasoning tasks.

**7. Limitations**

* Only **one model (gpt-4o)** has been evaluated.
* Dataset size is **relatively small** (10 examples). (Each example was repeated 3 times)
* The domain is narrow (model validation for risk/compliance) and may not generalize.

**8. Comparison with Prior Work**

Compared to the findings in the *Prompt Report* survey (arXiv:2307.05230), our results are aligned in several key ways:

* Few-shot and CoT prompting consistently outperform zero-shot prompting.
* Structured prompts like ReAct and ToT add reasoning and self-correction, which improves overall quality.
* Like the survey, we observe diminishing returns in performance gains when moving from CoT to even more complex prompting.

Our study is unique in focusing on **document validation tasks** within a regulatory setting—an area underrepresented in prior benchmarking studies.

**9. Remaining Tasks**

* Complete the remaining experiments using **LLaMA 3**, **Qwen 7B**
* Expand the dataset with more instructions and diverse context types
* Evaluate cost-effectiveness and latency of different techniques
* Automate metric scoring and error classification