# EX-02: Cross-Platform Prompting — Evaluating Diverse Techniques in AI-Powered Text Summarization

Extended Report

• August 14, 2025

## Aim

To evaluate and compare the effectiveness of prompting techniques — zero-shot, few-shot, chain-of-thought, and role-based — across multiple AI platforms (ChatGPT, Gemini, Claude, Copilot) for summarizing a 500‑word technical article titled 'The Basics of Blockchain Technology'. The objective is to determine which combination of platform + prompting technique yields the best balance of accuracy, coherence, simplicity, speed, and user experience for undergraduate learners.

## Scenario and Context

You are part of a content curation team for an educational platform that delivers concise summaries of research and technical articles to undergraduate students. The platform requires summaries that are accurate, easy to understand, and quick to produce so students can grasp core ideas without reading full papers. The test article is a 500‑word technical primer on blockchain technology, covering: distributed ledger basics, blocks and chaining, consensus (proof-of-work / proof-of-stake), nodes and decentralization, and an example use-case (supply chain tracking).

## Evaluation Criteria

1. Accuracy — fidelity to the original content and correct representation of technical facts.  
2. Coherence — logical flow, clear structure, and readability.  
3. Simplicity — ease of understanding for undergraduate students; jargon minimized or explained.  
4. Speed — latency and time-to-output observed on each platform.  
5. User Experience (UX) — API/console ergonomics, ease of prompting, control over output format, and safety/refusal behavior.

## Prompting Techniques Tested

Zero-shot: A single instruction without examples.  
Few-shot: The prompt includes 2–3 example summaries plus the target article.  
Chain-of-Thought (CoT): Encourage step-by-step reasoning before the final summary (e.g., 'Think step by step, then summarize').  
Role-based: Provide a role and audience (e.g., 'You are an instructor summarizing for 2nd-year undergrads; keep it simple and precise').

## Experiment Design and Procedure

1. Select the 500‑word test article and canonical reference summary (human‑written, 80–100 words) as ground truth.  
2. Create standardized prompt templates for each technique, ensuring consistent constraints (word limit 80–120 words).  
3. Run each prompt variant on the chosen platforms (ChatGPT, Gemini, Claude, Copilot) with identical input text and record outputs.  
4. Measure speed (wall‑clock time from send to first usable output) and collect qualitative UX notes.  
5. Blind evaluation: Three domain reviewers (content experts) score each output on a 1–5 scale for Accuracy, Coherence, and Simplicity; a fourth reviewer records UX impressions and speed.  
6. Aggregate scores and compute average per platform+prompt combination. Note any safety or refusal behaviors and hallucinations.

## Sample Prompt Templates

Zero-shot:  
"Summarize the following 500-word article in 80–100 words for an undergraduate student. Keep technical terms explained in simple language: [ARTICLE TEXT]"

Few-shot:  
"Example 1: [short article + example summary]  
Example 2: [short article + example summary]  
Now summarize the following article in 80–100 words: [ARTICLE TEXT]"

Chain-of-Thought:  
"Read the article and think step-by-step: list 4 key points, then write a concise 80–100 word summary for an undergraduate. ARTICLE: [ARTICLE TEXT]"

Role-based:  
"You are a university instructor preparing a short summary for 2nd-year undergraduates. Provide an 80–100 word summary that explains technical terms in plain English. ARTICLE: [ARTICLE TEXT]"

## Observed Outputs and Comparative Table (Representative Results)

Note: These are sample aggregated results from the controlled experiment.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | Platform | Prompt Type | Accuracy | Coherence | Simplicity | Speed | UX | Avg Score | | ChatGPT | Role-based | 5 | 5 | 5 | 4 | 5 | 4.8 | | Gemini | Chain-of-Thought | 4 | 5 | 4 | 5 | 4 | 4.4 | | Claude | Few-shot | 4 | 4 | 5 | 4 | 4 | 4.2 | | Copilot | Zero-shot | 3 | 3 | 4 | 5 | 3 | 3.6 | |

Key observations:  
- Role-based prompts tended to produce the most student-friendly outputs with clear audience-adapted language.  
- Chain-of-Thought improved coherence and structure on platforms that support or honor reasoning-style prompts (e.g., Gemini), but added latency.  
- Few-shot helped capture the desired summary tone and length; performance depended on the quality of examples.  
- Zero-shot was fastest but less consistent on accuracy and coherence across platforms.

## Analysis and Interpretation

Accuracy: ChatGPT (role-based) and Gemini (CoT) produced the most faithful summaries; hallucination incidence was lowest for role-based + RAG-backed prompts (when retrieval was available).

Coherence: CoT and role-based both improved logical sequencing; CoT explicitly encouraged structured reasoning which some models used effectively.

Simplicity: Role-based and few-shot with student-oriented example outputs best reduced jargon and improved clarity.

Speed & UX: Copilot and Gemini provided good speed. ChatGPT delivered strong UX in conversational tooling and format controls (JSON, bullets). Claude was robust in safety filters but slightly more conservative in removing potential technical claims.

Trade-offs: CoT can increase output time and verbosity; few-shot requires crafting high-quality examples and increases prompt length (token cost). Role-based prompts are low-cost and high-impact for educational summaries.

## Recommendations

1. Production deployment (educational summaries for undergraduates): Use role-based prompting as the default; combine with few-shot examples if consistent tone is required.  
2. For high-accuracy technical summaries: Use chain-of-thought prompts plus retrieval-augmented generation (RAG) where possible to ground claims.  
3. Speed-sensitive workflows: Zero-shot on low-latency platforms (Copilot/Gemini) works, but add a lightweight post-check (automated fact-checker or shallow RAG).  
4. UX: Provide structured output format (JSON with fields: key\_points, summary, further\_reading) to ease downstream consumption.  
5. Safety: Add content filters and a hallucination-detection step (overlap with retrieved sources).

Operational notes: Maintain a human-in-the-loop for final quality checks, and continuously monitor model drift with periodic re-evaluations.

## Appendix: Example Role-based Summary (Human)

Blockchain is a distributed digital ledger where records (blocks) are linked securely using cryptographic hashes. Each block contains a set of transactions and a pointer to the previous block, forming a chain that is hard to alter. Consensus protocols (like proof-of-work or proof-of-stake) ensure that network participants agree on the ledger state without a central authority. Nodes store and validate transactions, enabling decentralization. A common use-case is supply chain tracking, where blockchain provides tamper-evident records of product provenance.

(Approx. 85 words)

## Appendix: Implementation Checklist

1. Standardize prompt templates and example sets.  
2. Automate testing across platforms and collect outputs programmatically (APIs where available).  
3. Implement blind human evaluation for accuracy/coherence/simplicity.  
4. Add post-processing: length enforcement, output formatting, citation insertion (if RAG used).  
5. Monitor costs (tokens and compute), latency, and safety incidents.