1. Prompt Templates

This document collects all prompt templates used in the agentic RAG (A-RAG) and agentic SFT (A-SFT) frameworks.

1.1. A-RAG Prompts

1.1.1. Initial Detection Prompt Template

Inputs: `$CODE_SNIPPET`.

Prompt: "You are an expert static-analysis agent. Analyze the following C code snippet and determine whether it contains a fault ($CLASSIFICATION_LABEL = 1) or not ($CLASSIFICATION_LABEL = 0). Provide a concise explanation citing the specific operations, expressions, or control-flow behaviors that determined your decision."

Outputs: `$CLASSIFICATION_LABEL`, `$REASONING`.

1.1.2. Confidence Estimation Prompt Template

Inputs: `$CODE_SNIPPET`, `$CLASSIFICATION_LABEL`, `$REASONING`.

"Based on the reasoning you just provided, output a single number in $[0, 1]$ representing your confidence in the correctness of your classification of the code snippet. Output only the number."

Outputs: `$CONFIDENCE_SCORE`.

1.1.3. Adaptive Query Prompt Template

Inputs: `$CODE_SNIPPET`, `$CLASSIFICATION_LABEL`, `$REASONING`, `$CONFIDENCE_SCORE`.

"Use the previous reasoning to generate a precise search query for C-language fault patterns. Identify the operations, variables, conditions, or memory-access patterns responsible for uncertainty and transform them into a structured, fault-oriented query suitable for retrieving relevant CWE and SEI CERT guidance."

Outputs: `$RETRIEVAL_QUERY`.

1.1.4. Context-Augmented Detection Prompt Template

Inputs: `$CODE_SNIPPET`, `$RETRIEVED_CONTEXT`.

"Reassess the following C code snippet using the retrieved CWE/SEI knowledge. Incorporate the provided contextual information to refine your classification. Explain whether the snippet is faulty ($CLASSIFICATION_LABEL = 1) or non-faulty ($CLASSIFICATION_LABEL = 0), explicitly referencing relevant fault patterns, unsafe behaviors, or remediation principles from the retrieved knowledge base."

Outputs: `$CLASSIFICATION_LABEL`, `$REASONING`.

1.2. A-SFT Prompts

1.2.1. Self-Evaluation Prompt Template

Inputs: `$CODE_SNIPPET`, `$INITIAL_LABEL`, `$REASONING`, `$CONFIDENCE_SCORE`.

"Evaluate your previous answer. Identify any reasoning flaws, missing considerations, or incorrect assumptions. If the classification was incorrect or uncertain, propose a corrected classification and provide a concise justification referencing the specific memory, numerical, control-flow, or concurrency behavior involved."

Outputs: `$SELF_CRITIQUE`, `$REVISED_LABEL`, `$REVISED_REASONING`.

1.2.2. Instruction Adaptation Prompt Template

Inputs: `$SELF_CRITIQUE_SUMMARIES`.

"Identify recurring reasoning errors across all evaluated samples. Rewrite or extend the system instruction to explicitly address these weaknesses. Provide updated guidance, decision rules, or reasoning heuristics that will help avoid similar mistakes in the future."

Outputs: `$UPDATED_SYSTEM_INSTRUCTION`.