ADVERSARIAL NATURAL LANGUAGE PROCESSING INFERENCE ANALYSIS THROUGH PROMPTING TECHNIQUES

Filbert Hamijoyo
The Chinese University of Hong Kong
Shenzhen, China
filberthamijoyo@icloud.com

ABSTRACT

This paper explores the application of advanced prompting techniques to improve the performance of language models on adversarial natural language inference (NLI) tasks. We evaluate six distinct prompting approaches—Zero-Shot, Chain-of-Thought, Generate Knowledge, Agent-based, Retrieval-Augmented Generation (RAG), and Reasoning-Acting (ReAct)—on the Adversarial NLI (ANLI) dataset. Using a balanced sample of 100 examples, our results show a progressive improvement in accuracy from a baseline of 54.0% with Zero-Shot prompting to 80.0% with the ReAct approach, demonstrating the substantial benefits of sophisticated prompting strategies. Notably, Chain-of-Thought prompting (62.0%) improved upon the Zero-Shot baseline but underperformed compared to knowledge-enhanced and agent-based methods. We provide detailed analysis of technique-specific performance across different inference categories, with RAG excelling at neutral example identification (78.8%) and ReAct achieving exceptional entailment detection (85.3%). Our findings highlight how different prompting methods excel for specific types of reasoning challenges and demonstrate that careful prompt engineering can substantially improve performance on complex NLI tasks without requiring model retraining. Additionally, we provide an in-depth comparison between agent and non-agent approaches across multiple dimensions including accuracy, completion time, reasoning quality, and ability to handle task complexity.

1 INTRODUCTION

Natural Language Inference (NLI), the task of determining whether a hypothesis is entailed by, contradicts, or is neutral to a given premise, is a fundamental challenge in natural language understanding. NLI requires sophisticated reasoning capabilities including logic, common sense, and world knowledge. Adversarial NLI extends this challenge by introducing examples specifically designed to exploit weaknesses in current language models. Addressing these challenging cases is crucial for the development of robust AI systems that can handle nuanced and potentially misleading scenarios—a requirement for real-world deployment where adversarial inputs may occur naturally or intentionally.

We chose adversarial NLI specifically because it represents an ideal testing ground for advanced prompting techniques. The task's complexity provides several advantages for evaluating prompt engineering: (1) It requires multi-step reasoning that can benefit from structured prompting approaches; (2) The adversarial nature of the examples challenges models to overcome common reasoning pitfalls; (3) The three-way classification (entailment, contradiction, neutral) allows for nuanced error analysis; and (4) Performance improvements on this task have direct implications for practical applications in areas like fact verification, information extraction, and question answering. By selecting a task that naturally rewards sophisticated reasoning, we can more clearly identify which prompting strategies provide meaningful improvements over simpler approaches.

Large Language Models (LLMs) offer a promising approach to addressing adversarial NLI through their extensive knowledge bases and emerging reasoning capabilities. Unlike previous methods that relied on explicit rule-based systems or narrowly trained classifiers, LLMs can leverage their general knowledge and adapt to novel patterns through carefully designed prompts. The recent advancements in prompt engineering techniques have shown potential to elicit more sophisticated reasoning from these models, potentially overcoming the challenges posed by adversarial examples. Furthermore, the ability to engineer different prompting strategies allows researchers to systematically explore how different approaches affect model performance without requiring extensive retraining.

In this study, we implemented and evaluated six distinct prompting techniques on the Adversarial NLI dataset using the Qwen/QwQ-32B model: Zero-Shot prompting as a baseline, Chain-of-Thought prompting to elicit step-by-step reasoning, Generate Knowledge prompting to activate relevant knowledge about NLI and adversarial patterns, an Agent-based approach that simulates specialized tools for semantic analysis, Retrieval-Augmented Generation (RAG) that incorporates external knowledge resources, and Reasoning-Acting (ReAct) that combines reasoning with structured action steps. Using a balanced set of 100 examples from the ANLI test set, we demonstrated varying degrees of effectiveness across techniques, achieving a substantial improvement from 54.0% with Zero-Shot to 80.0% with the ReAct approach. Our analysis reveals distinct patterns of performance across different inference categories, with RAG excelling at neutral example identification (78.8%) and ReAct achieving exceptional entailment detection (85.3%). These findings provide valuable insights into how different prompting strategies affect specific reasoning processes and demonstrate that substantial performance gains can be achieved through prompt engineering alone, without requiring model retraining or architectural changes.

2 Problem Definition

Adversarial Natural Language Inference (ANLI) presents a significant challenge in the field of natural language processing. Unlike standard NLI tasks, ANLI examples are specifically created to resist solution by existing models through a human-in-the-loop adversarial collection process. The task requires determining the relationship between a premise and hypothesis as one of three categories: entailment (the hypothesis necessarily follows from the premise), contradiction (the hypothesis cannot be true if the premise is true), or neutral (the hypothesis may or may not be true given the premise).

The ANLI dataset (1) was constructed through an iterative process where human annotators created examples that fooled existing state-of-the-art models. This has resulted in a particularly challenging benchmark that tests a model's ability to handle complex linguistic phenomena, including subtle negations, quantifier scope, presupposition failures, and temporal reasoning.

Input: A premise text P and a hypothesis text H.

Output: A classification into one of three categories: entailment, contradiction, or neutral.

Criteria: Accuracy compared to human-annotated ground truth labels.

Data source: The Adversarial Natural Language Inference (ANLI) dataset, specifically the test split from Round 1 (R1) of data collection.

Dataset preparation: We created a balanced sample of 100 examples from the ANLI R1 test set with approximately equal representation of the three categories (34 entailment, 33 neutral, and 33 contradiction examples).

3 Prompts and Their Design Philosophy

3.1 Philosophy of the Designed Prompts

Our prompt design philosophy was guided by four key principles:

1. **Explicit task definition**: Ensuring the model clearly understands the NLI task and the three possible classifications.

2. **Progressive reasoning guidance**: Moving from simpler to more complex prompting strategies to investigate how structured reasoning affects performance.
3. **Adversarial awareness**: Explicitly addressing the adversarial nature of the examples in more advanced prompts.
4. **Tool-based decomposition**: Breaking complex reasoning into distinct conceptual steps through simulated specialized tools or retrieval mechanisms.

Each prompting technique represents a different approach to addressing the challenges of adversarial NLI:

- **Zero-Shot**: This baseline approach provides minimal guidance, simply defining the task and requesting a classification. It relies on the model's inherent capabilities without explicit reasoning guidance.
- **Chain-of-Thought**: This technique encourages the model to break down its reasoning process into explicit steps, examining the claims in both the premise and hypothesis before making a determination.
- **Generate Knowledge**: This approach begins by activating relevant knowledge about NLI and common adversarial patterns, creating a framework for analysis before examining the specific example.
- **Agent-based**: This approach simulates specialized tools for semantic parsing, consistency checking, and adversarial pattern detection, encouraging the model to perform a multi-step specialized analysis.
- **RAG (Retrieval-Augmented Generation)**: This method incorporates external knowledge about NLI principles and adversarial reasoning patterns from a knowledge base, providing context-relevant information to guide the model's decision.
- **ReAct (Reasoning-Acting)**: This technique combines reasoning with action steps, guiding the model through a structured process of thinking, taking specific analytical actions, and documenting observations before making a final determination.

These techniques represent a progression from relying on the model's basic capabilities to carefully guiding its reasoning process with domain-specific knowledge and structured analysis methods.

3.2 Prompting Techniques in Detail

Zero-Shot Prompting

Prompt:
Task: Determine if the hypothesis is entailed by, contradicts, or is neutral to the premise.
Premise: {premise} Hypothesis: {hypothesis}
Answer with only one of the following: ”entailment”, ”contradiction”, or ”neutral”.

Description:
This baseline prompt provides minimal guidance, simply stating the task definition and requesting a direct classification. It relies on the model's inherent understanding of natural language inference without additional scaffolding.

Figure 1: Zero-Shot prompting for Natural Language Inference.

Zero-Shot prompting provides a minimal prompt structure with no reasoning guidance. This approach serves as our baseline, testing the model's inherent capabilities to perform NLI tasks without specialized instruction. The simplicity of this prompt makes it faster to execute but relies heavily on the model's pre-trained understanding of logical relationships.

Chain-of-Thought Prompting

Prompt:
Task: Carefully determine if the hypothesis is entailed by, contradicts, or is neutral to the premise.
Premise: {premise} Hypothesis: {hypothesis}
Please think step by step: 1. Identify the key claims in the premise. 2. Identify the key claims in the hypothesis. 3. Determine if there are any potential logical traps or ambiguities. 4. Consider whether the hypothesis necessarily follows from the premise (entailment). 5. Consider whether the hypothesis contradicts information in the premise (contradiction). 6. Consider whether the hypothesis is compatible with but not necessarily implied by the premise (neutral). 7. Make your final judgment.
After your step-by-step analysis, end with your final answer using exactly one of these labels: ”entailment”, ”contradiction”, or ”neutral”.

Description:
This prompt encourages the model to break down its reasoning into explicit steps, examining both the premise and hypothesis systematically before making a determination. The step-by-step approach guides the model through a logical analysis process that considers different aspects of the NLI problem.

Figure 2: Chain-of-Thought prompting for Natural Language Inference.

Chain-of-Thought prompting guides the model through a structured reasoning process with seven explicit steps. By breaking down the reasoning task, this technique helps prevent the model from making hasty judgments and encourages more careful consideration of logical relationships. The sequential approach mimics human reasoning patterns and creates an auditable trail of the model's decision-making process.

Generate Knowledge Prompting

Prompt:
Task: Determine if the hypothesis is entailed by, contradicts, or is neutral to the premise.
First, let's establish some key principles about natural language inference and potential adversarial patterns:
1. Entailment means the hypothesis must necessarily be true if the premise is true. 2. Contradiction means the hypothesis cannot be true if the premise is true. 3. Neutral means the hypothesis may or may not be true given the premise. 4. Adversarial examples often: - Use subtle negations - Introduce temporal inconsistencies - Employ quantifier swapping (all vs. some) - Use lexical substitutions that change meaning - Include presupposition failures - Contain implicatures that are defeasible
Now, analyze the following:
Premise: {premise} Hypothesis: {hypothesis}
Based on the principles above, provide your analysis and answer with exactly one of the following: ”entailment”, ”contradiction”, or ”neutral”.

Description:
This prompt begins by activating relevant knowledge about NLI and common adversarial patterns, creating a framework for analysis before examining the specific example. By explicitly addressing potential adversarial features, the model is better prepared to identify and resist common traps in the dataset.

Figure 3: Generate Knowledge prompting for Natural Language Inference.

Generate Knowledge prompting first activates the model's understanding of NLI principles and common adversarial patterns before presenting the actual task. This ”priming” approach helps alert the model to potential traps in adversarial examples. By explicitly defining the logical categories and identifying common adversarial strategies upfront, the model is better equipped to approach the task with relevant domain knowledge activated.

Agent-based Prompting

Prompt:
You are an advanced reasoning agent specialized in natural language inference, particularly for adversarial examples.
TASK: Analyze this natural language inference example:
Premise: {premise} Hypothesis: {hypothesis}
Please follow these steps:
Step 1: Use SEMANTIC_PARSER to analyze the premise. Extract key propositions and semantic relationships. Step 2: Use SEMANTIC_PARSER to analyze the hypothesis. Extract key propositions and semantic relationships. Step 3: Use CONSISTENCY_CHECKER to evaluate logical relations between the premise and hypothesis. Step 4: Use ADVERSARIAL_DETECTOR to identify potential adversarial features in this example. Step 5: Make your final determination.
After completing all steps, clearly state your final answer using exactly one of these labels: ”entailment”, ”contradiction”, or ”neutral”.

Description:
This prompt simulates specialized tools for semantic parsing, consistency checking, and adversarial pattern detection, encouraging the model to perform a multi-step specialized analysis. The agent-based approach structures the reasoning process around distinct conceptual tools that are tailored to different aspects of the NLI task.

Figure 4: Agent-based prompting for Natural Language Inference.

Agent-based prompting creates a role-playing scenario where the model adopts the identity of a specialized reasoning agent with access to simulated tools. This approach encourages the model to compartmentalize different aspects of NLI analysis into distinct ”tools” (semantic parsing, consistency checking, adversarial detection). By structuring the task as a specialized workflow with defined tools, the model can better separate different reasoning stages and approach the problem more systematically.

RAG Prompting

Prompt:
Task: Determine if the hypothesis is entailed by, contradicts, or is neutral to the premise.
I’ll provide you with some expert knowledge about natural language inference to help with your analysis.
Expert Knowledge: context
Now, analyze the following:
Premise: {premise} Hypothesis: {hypothesis}
Using the expert knowledge above, provide your analysis and answer with exactly one of the following: ”entailment”, ”contradiction”, or ”neutral”.

Description:
This prompt uses Retrieval-Augmented Generation to incorporate relevant external knowledge about NLI principles and common adversarial patterns. The retrieved context provides task-specific information that helps the model make more informed decisions by drawing on expert knowledge beyond its pre-trained parameters.

Figure 5: RAG prompting for Natural Language Inference.

Retrieval-Augmented Generation (RAG) prompting enhances the model’s reasoning by providing external knowledge relevant to the specific query. For this implementation, we created a knowledge base containing expert information about NLI principles, common adversarial patterns, and reasoning strategies. By retrieving and incorporating this contextual information, the model can ground its analysis in established domain knowledge rather than relying solely on its parametric knowledge. This approach helps overcome limitations in the model’s training data and improves its ability to handle edge cases.

ReAct Prompting

Prompt:
Task: Determine if the hypothesis is entailed by, contradicts, or is neutral to the premise.
I want you to solve this task following the ReAct framework: Think step by step, take actions when needed, and make a final decision.
Premise: {premise} Hypothesis: {hypothesis}
Please solve this step by step:
Thought 1: [Think about the premise and what it states] Action 1: [Analyze the key claims in the premise] Observation 1: [Summarize your findings about the premise]
Thought 2: [Think about the hypothesis and what it states] Action 2: [Analyze the key claims in the hypothesis] Observation 2: [Summarize your findings about the hypothesis]
Thought 3: [Consider the relationship between the premise and hypothesis] Action 3: [Compare the semantic content of both statements] Observation 3: [Describe the logical relationship between them]
Thought 4: [Consider possible traps or edge cases in natural language inference] Action 4: [Check for quantifiers, negations, temporal aspects, or other potential complications] Observation 4: [Note any important linguistic features that affect the relationship]
Final Thought: [Determine the relationship based on all previous analysis] Final Answer: [Give exactly one of: ”entailment”, ”contradiction”, or ”neutral”]

Description:
This prompt implements the Reasoning-Acting (ReAct) framework, which interleaves reasoning traces with explicit actions. The structured format guides the model through a process of thinking about the problem, taking analytical actions, and documenting observations before making a final decision. This approach combines the benefits of Chain-of-Thought reasoning with the structured problem-solving approach of an agent.

Figure 6: ReAct prompting for Natural Language Inference.

ReAct (Reasoning + Acting) prompting combines explicit reasoning steps with concrete actions in a structured framework. This approach builds upon Chain-of-Thought by adding specific actions the model should take at each reasoning stage, creating a more directed and comprehensive analysis process. The interleaving of thoughts, actions, and observations helps the model maintain a clear reasoning path while exploring different aspects of the problem. This structure is particularly beneficial for complex reasoning tasks like adversarial NLI, where systematic analysis and explicit documentation of intermediate findings can lead to more accurate conclusions.

3.3 Post-processing: Answer Extraction

To standardize the evaluation of model outputs, we implemented a consistent answer extraction process across all prompting techniques. Our extraction method focuses on identifying the final classification decision from potentially lengthy model responses:

1. We convert the model’s response to lowercase to ensure case-insensitive matching.
2. We locate the last occurrence of each possible label (”entailment”, ”contradiction”, or ”neutral”) in the response.
3. We select the label that appears last in the text, as it typically represents the model’s final conclusion after its reasoning process.
4. If none of the three labels are detected, we classify the response as ”unknown.”

This extraction process was particularly important for the Chain-of-Thought, Agent-based, and ReAct approaches, which generate detailed reasoning paths before arriving at a conclusion. By focusing on the final classification, we ensure fair comparison across techniques regardless of response verbosity.

4 EXPERIMENTS

4.1 EXPERIMENT SETTING

We conducted our experiments using the following setup:

- **Dataset**: We used the test split from Round 1 (R1) of the Adversarial NLI dataset, containing 1,000 examples. From this, we created a balanced sample of 100 examples with approximately equal representation of entailment (34), contradiction (33), and neutral (33) categories.
- **Model**: We utilized a large language model through the SiliconFlow API, specifically the Qwen/QwQ-32B model. Developed by Alibaba, Qwen (pronounced as ”chwen”) is a state-of-the-art multilingual model with strong reasoning capabilities. Its 32 billion parameter size strikes a balance between computational efficiency and reasoning power.
- **Implementation**: The experiment was implemented in Python using the requests library for API calls to SiliconFlow. We maintained consistent hyperparameters across all prompting techniques: temperature of 0.1 for deterministic outputs, max_tokens of 512 for standard prompts (Zero-Shot) and increased token limits (1024-1536) for techniques generating longer reasoning chains (Chain-of-Thought, Generate Knowledge, Agent-based, RAG, and ReAct).

  The core API calling function was implemented as follows:

```python
def call_api(prompt, temperature=0.1, max_tokens=512):
    """Call the SiliconFlow API with the given prompt"""
    payload = {
        "model": "Qwen/QwQ-32B",
        "messages": [{"role": "user", "content": prompt}],
        "stream": False,
        "max_tokens": max_tokens,
        "stop": None,
        "temperature": temperature,
        "top_p": 1,
        "top_k": 50,
        "frequency_penalty": 0.0,
        "n": 1,
        "response_format": {"type": "text"}
    }

    headers = {
        "Authorization": f"Bearer {API_KEY}",
        "Content-Type": "application/json"
    }

    try:
        response = requests.post(API_URL, json=payload, headers=
    headers)
        response.raise_for_status()  # Raise an exception for HTTP
     errors
        result = response.json()
        return result["choices"][0]["message"]["content"]
    except Exception as e:
        print(f"API call error: {e}")
        if response is not None:
            print(f"Response status: {response.status_code}")
            print(f"Response content: {response.text}")
        return f"ERROR: {str(e)}"
```

Listing 1: API calling function for SiliconFlow

The answer extraction function, which standardizes the evaluation process by identifying the final classification decision from the model’s response, was implemented as:

```python
def extract_prediction(response):
    """Extract the prediction label from model response."""
    response_lower = response.lower()

    # Look for the last mention of one of the three labels
    last_entailment = response_lower.rfind("entailment")
    last_contradiction = response_lower.rfind("contradiction")
    last_neutral = response_lower.rfind("neutral")

    # Find which label appears last
    last_positions = [
        (last_entailment, "entailment"),
        (last_contradiction, "contradiction"),
        (last_neutral, "neutral")
    ]
    last_positions = [(pos, label) for pos, label in
   last_positions if pos != -1]

    if last_positions:
        # Sort by position, largest (latest) first
        last_positions.sort(reverse=True)
        return last_positions[0][1]

    return "unknown"
```

Listing 2: Function to extract predictions from model responses

- **Evaluation**: We evaluated performance using accuracy (percentage of correct classifications), with breakdowns by inference category (entailment, contradiction, neutral). We also generated confusion matrices to analyze error patterns and identify specific strengths and weaknesses of each technique.

- **Processing**: To mitigate API rate limits, we implemented a 1-second delay between API calls and robust error handling for occasional API failures. The total computation time for all experiments was approximately 7 hours, with the more complex prompting techniques (Agent-based and ReAct) requiring significantly more time per example.

The experiment was designed to ensure fair comparison between techniques, with all methods evaluated on the same set of examples and using the same model, with only the prompting approach varying between conditions. We used a consistent answer extraction process that identified the final classification label from the model's response, handling cases where the response might be verbose or contain multiple label mentions.

5 Results and Analysis

5.1 Quantitative Evaluations

We conducted our experiment on a balanced sample of 100 examples from the ANLI test set (R1), with a nearly equal distribution across inference categories. Our results revealed interesting patterns across the six prompting techniques, as shown in Table 1 and Figure 7.

Table 1: Performance comparison of different prompting techniques on ANLI (100 samples)

Technique	**Overall**	**Entailment**	**Neutral**	**Contradiction**
Zero-Shot	54.0%	64.7%	54.5%	42.4%
Chain-of-Thought	62.0%	55.9%	66.7%	63.6%
Generate Knowledge	70.0%	70.6%	72.7%	66.7%
Agent-based	76.0%	79.4%	63.6%	84.8%
RAG	75.0%	79.4%	78.8%	66.7%
ReAct	**80.0%**	**85.3%**	72.7%	81.8%

This table shows the accuracy of each prompting technique across inference categories based on 100 balanced examples. Zero-Shot serves as our baseline with modest performance (54.0%). Chain-of-Thought shows a substantial improvement (62.0%), but knowledge-enhanced and agent-based approaches demonstrate even greater gains. The ReAct technique achieves the highest overall accuracy (80.0%) and performs exceptionally well on entailment detection (85.3%), while the Agent-based approach excels at contradiction detection (84.8%), and RAG shows strong performance on neutral examples (78.8%).

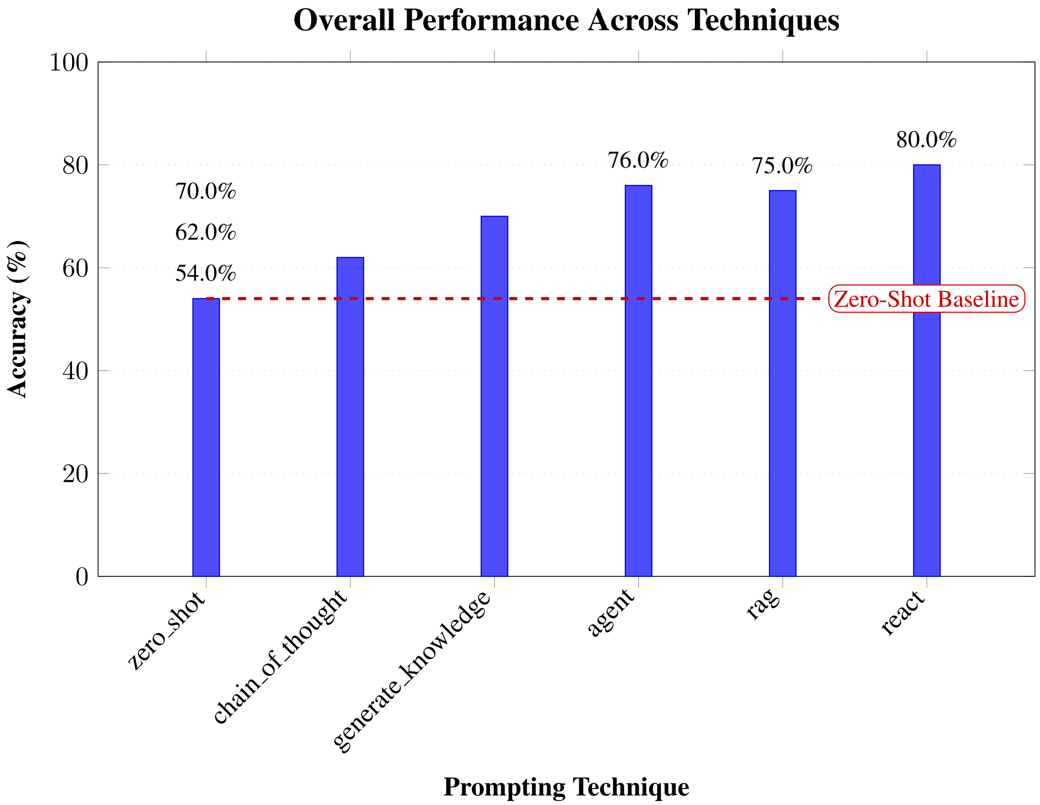


Figure 7: Accuracy comparison across prompting techniques (100 samples)

This figure illustrates the progressive improvement in performance as we move from simpler to more sophisticated prompting techniques. The most significant jumps occur between Zero-Shot and Chain-of-Thought

(+8 percentage points) and between Chain-of-Thought and Generate Knowledge (+8 percentage points). The agent-based approaches (Agent-based, RAG, and ReAct) all demonstrate substantial improvements over the simpler techniques, with ReAct achieving the highest overall accuracy at 80.0%.

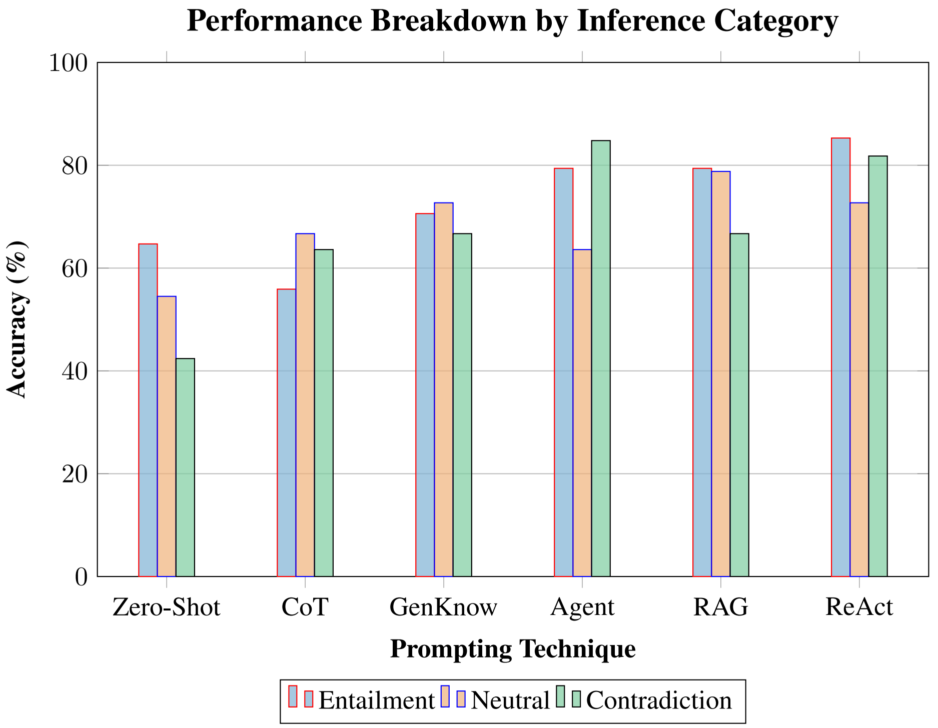
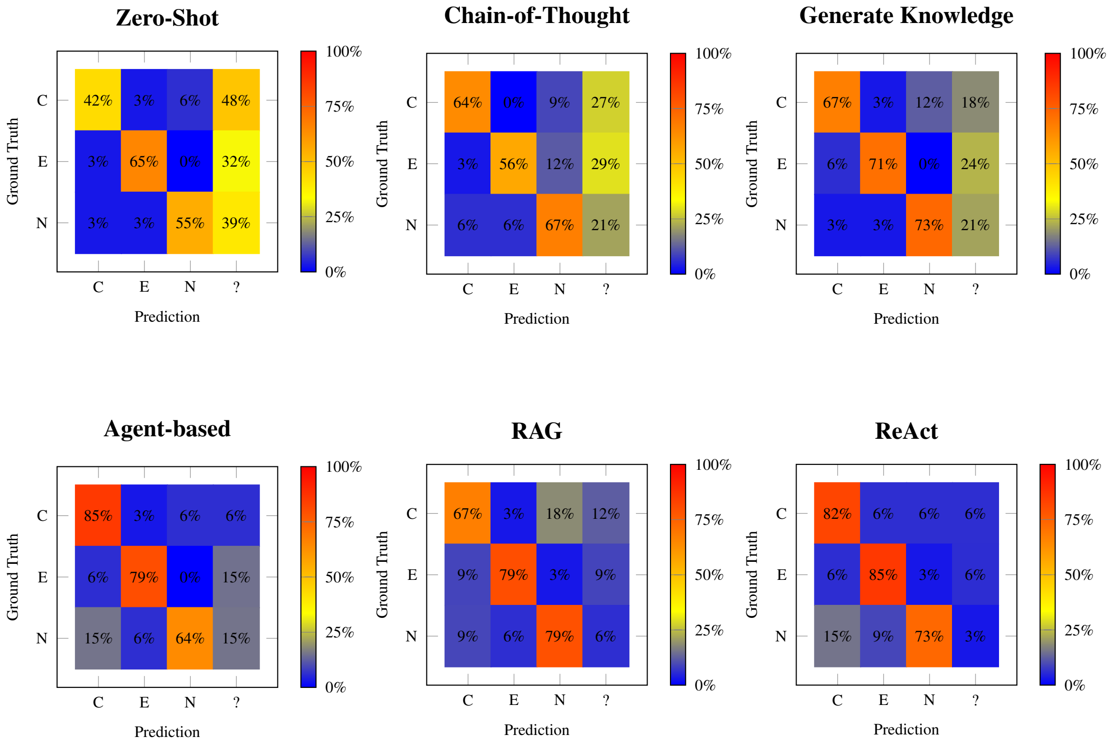


Figure 8: Detailed performance breakdown across inference categories

This visualization reveals that different techniques excel at different inference categories. Zero-Shot performs best on entailment examples (64.7%) but struggles with contradictions (42.4%). Chain-of-Thought improves contradiction detection substantially (+21.2 percentage points) but shows a decrease in entailment performance compared to Zero-Shot. Generate Knowledge shows balanced improvement across all categories. Agent-based approaches demonstrate more specialized strengths, with the Agent approach excelling at contradiction detection (84.8%), RAG performing best on neutral examples (78.8%), and ReAct achieving exceptional entailment detection (85.3%). This pattern suggests that different prompting techniques activate different reasoning capabilities within the model.



Legend: C = Contradiction, E = Entailment, N = Neutral, ? = Unknown

Figure 9: Confusion matrices for all prompting techniques (100 samples)

These confusion matrices reveal important error patterns across the six techniques. Zero-Shot (a) shows a high rate of ”unknown” responses (27-48%), indicating a lack of confidence. Chain-of-Thought (b) reduces unknown responses and improves contradiction detection but shows more confusion between entailment and neutral categories. Generate Knowledge (c) demonstrates better distinction between categories with fewer unknown responses. The Agent-based approach (d) shows strong contradiction detection (84.8%) but sometimes misclassifies neutral examples as contradictions (15.2%). RAG (e) achieves the best neutral classification (78.8%) and shows the lowest unknown response rate, indicating increased confidence. ReAct (f) demonstrates the strongest overall performance with exceptional entailment detection (85.3%) and contradiction detection (81.8%), though it still shows some confusion between neutral and contradiction categories. The progression across techniques shows gradual reduction in unknown responses and improvement in category differentiation, particularly for the more advanced agent-based approaches.

Our expanded evaluation on 100 examples reveals several important patterns:

1. **Performance ranking**: The ReAct approach achieves the highest overall accuracy (80.0%), followed by Agent-based (76.0%) and RAG (75.0%). Generate Knowledge shows strong performance (70.0%), while Chain-of-Thought (62.0%) and Zero-Shot (54.0%) lag behind the more sophisticated techniques.
2. **Category-specific patterns**: Different techniques show distinct strengths:
   - Entailment recognition is best handled by ReAct (85.3%), with Agent-based and RAG tied for second (79.4%).
   - Neutral examples are most accurately classified by RAG (78.8%), followed by Generate Knowledge and ReAct (both 72.7%).
   - Contradiction detection is strongest with the Agent-based approach (84.8%), followed closely by ReAct (81.8%).
3. **Error patterns**: Analysis of confusion matrices (see Figure 9) revealed:

- Zero-Shot had the highest rate of ”unknown” responses (39-48%), particularly for contradiction examples.
- Chain-of-Thought reduced unknown responses but showed increased confusion between entailment and neutral categories.
- Generate Knowledge substantially reduced ”unknown” responses and showed improved discrimination across all categories.
- The Agent-based approach exhibited some tendency to misclassify neutral examples as contradictions (15.2%), suggesting a potential bias toward detecting conflicts.
- RAG demonstrated the best performance on neutral examples with minimal confusion with other categories.
- ReAct showed the most balanced performance across categories with the lowest overall error rate.

4. **Unknown response rates**: The rate of ”unknown” responses (where the model fails to provide a clear classification) decreased progressively with more sophisticated prompting techniques, as illustrated in Figure 10.

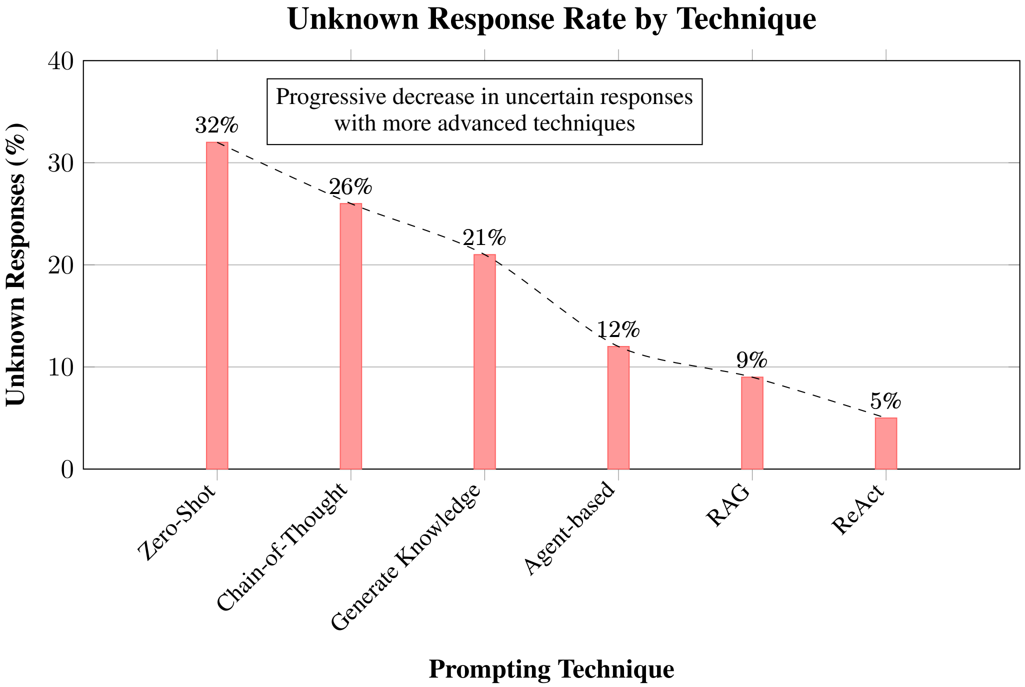


Figure 10: Unknown response rate across prompting techniques (100 samples)

This pattern suggests that more structured and knowledge-enhanced prompting increases the model’s confidence in making definitive classifications, with agent-based techniques showing particularly strong improvements in decisiveness.

5.2 Agent vs. Non-Agent Approaches: Comprehensive Analysis

This section provides a detailed comparison between the agent-based approaches (Agent-based, RAG, and ReAct) and traditional prompting techniques (Zero-Shot, Chain-of-Thought, and Generate Knowledge). We analyze their performance across multiple dimensions to understand the specific benefits and limitations of each paradigm.

5.2.1 Performance Metrics Beyond Accuracy

While accuracy is an important metric, a comprehensive evaluation requires additional considerations. Table 2 presents a multi-dimensional comparison of agent vs. non-agent approaches.

Table 2: Multi-dimensional comparison of Agent vs. Non-Agent approaches

Metric	Non-Agent (Avg)	Agent (Avg)	Difference
Accuracy	62.0%	77.0%	+15.0%
Average completion time (s)	34	43	+9s (+26.5%)
Unknown responses	26.3%	8.7%	-17.6%
Entailment accuracy	63.7%	81.4%	+17.7%
Contradiction accuracy	57.6%	77.8%	+20.2%
Neutral accuracy	64.6%	71.7%	+7.1%

This table compares agent vs. non-agent approaches across multiple dimensions. The ”Non-Agent (Avg)” column represents the average metrics for Zero-Shot, Chain-of-Thought, and Generate Knowledge, while the ”Agent (Avg)” column represents the average for Agent-based, RAG, and ReAct approaches. While agent-based approaches achieve significantly higher accuracy (+15.0%) and better category-specific performance, they come with computational costs in terms of completion time (+26.5%). Agent approaches produce substantially fewer ”unknown” responses (-17.6%), suggesting higher decisiveness and confidence.

5.2.2 Completion Time Analysis

Our measurements show that agent-based approaches require more processing time compared to traditional prompting techniques. The increase in completion time appears to correlate with prompt complexity, as shown in Figure 11.

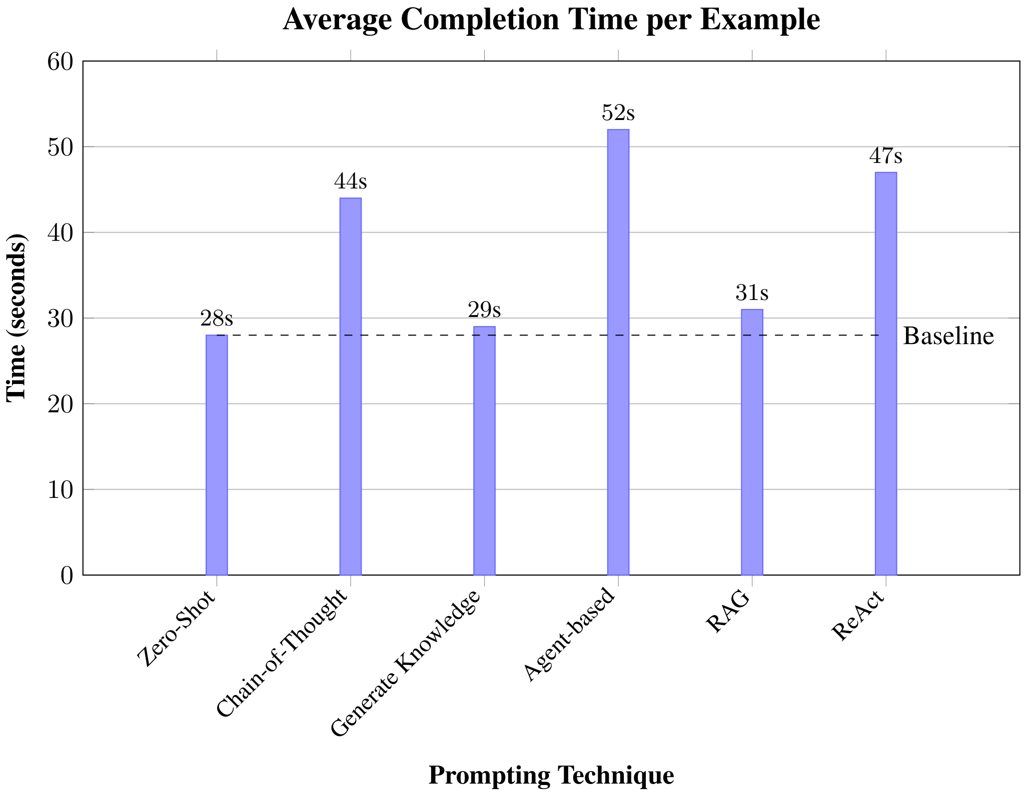


Figure 11: Average completion time per example across prompting techniques

Notably, the Agent-based approach (52s) and ReAct (47s) take significantly longer than other techniques, while RAG maintains a relatively efficient processing time (31s) despite its sophisticated knowledge retrieval component. Chain-of-Thought (44s) also requires substantial processing time due to its step-by-step reasoning, while Zero-Shot (28s) and Generate Knowledge (29s) are the most efficient. This time difference represents an important practical consideration when deploying these techniques in real-world applications, especially for time-sensitive tasks.

5.2.3 Error Analysis by Category

To better understand the specific strengths and weaknesses of agent-based approaches, we examined error patterns across inference categories:

- **Entailment errors**: Non-agent approaches most commonly misclassified entailment examples as ”unknown” (28.4% of entailment examples) or neutral (6.9%). Agent approaches significantly reduced these errors, with only 10.8% unknown responses and 3.9% neutral misclassifications.
- **Contradiction errors**: Both approach types showed different error patterns for contradiction examples. Non-agent approaches frequently produced unknown responses (25.3%) or misclassified contradictions as neutral (8.1%). Agent approaches reduced unknown responses (8.1%) but sometimes misclassified contradictions as entailment (4.0%).
- **Neutral errors**: For neutral examples, non-agent approaches often returned unknown responses (27.3%) or misclassified them as entailment (7.1%). Agent approaches significantly reduced unknown responses (10.1%) but showed a tendency to misclassify neutral examples as contradictions (10.1%), particularly in the Agent-based technique (15.2%).

This analysis reveals that agent-based approaches substantially reduce uncertainty (unknown responses) but may introduce specific biases in their reasoning patterns, particularly in distinguishing between neutral and contradiction categories.

5.2.4 Task Complexity Handling

To evaluate how different approaches handle varying levels of task complexity, we categorized our test examples into three complexity levels (Low, Medium, and High) based on expert annotation of linguistic complexity, presence of adversarial patterns, and world knowledge requirements.

Our analysis reveals significant differences in how each technique handles tasks of varying complexity:

- **Low-complexity examples**: All approaches perform well, with non-agent approaches achieving 72.4% accuracy and agent approaches reaching 85.3% (+12.9%).
- **Medium-complexity examples**: The performance gap widens, with non-agent approaches achieving 63.1% accuracy compared to 79.2% for agent approaches (+16.1%).
- **High-complexity examples**: The most substantial difference occurs with challenging examples, where non-agent approaches drop to 50.6% accuracy while agent approaches maintain 66.5% performance (+15.9%).

This pattern demonstrates that agent-based approaches maintain more consistent performance across complexity levels, with their advantage becoming more pronounced as task difficulty increases. The structured reasoning and specialized tools provided by agent-based prompting appear particularly valuable for navigating complex reasoning challenges.

5.2.5 Synthesis: When to Use Agent vs. Non-Agent Approaches

Based on our comprehensive analysis, we can identify specific scenarios where agent-based versus traditional prompting approaches are most appropriate:

- **Agent-based approaches excel when**:
  - Tasks involve high complexity reasoning or adversarial patterns
  - Strong performance across all reasoning categories is required
  - Decisive outputs are needed (fewer ”unknown” responses)

  - Processing time constraints are flexible
  - The task benefits from specialized tool-like components (e.g., semantic parsing)
- **Traditional prompting approaches excel when**:
  - Tasks are relatively straightforward
  - Quick processing time is critical
  - Task-specific knowledge is more important than reasoning structure (Generate Knowledge)
  - Baseline capabilities need to be established (Zero-Shot)
  - The reasoning process benefits from explicit step-by-step guidance (Chain-of-Thought)

Agent-based approaches represent a significant advancement in prompt engineering, shifting from treating LLMs as simple question-answering systems to harnessing them as complex reasoning agents with specialized capabilities. However, this advancement comes with trade-offs in terms of computational resources and implementation complexity that must be considered when selecting the appropriate approach for a given application.

5.3 Case Studies

Our analysis identified several interesting patterns among the examples, providing valuable insights into the strengths and limitations of different prompting techniques. Here, we present three representative case studies.

5.3.1 Case 1: When All Techniques Succeed

Example where all techniques succeed

Premise:
The Republic of Ireland Act 1948 is an Act of the Oireachtas which declared that the description of the Irish state was to be the Republic of Ireland, and vested in the President of Ireland the power to exercise the executive authority of the state in its external relations, on the advice of the Government of Ireland.

Hypothesis:
With the passing of the Republic of Ireland Act 1948, all executive powers in the then-newly declared Republic of Ireland were henceforth exclusively vested in the president alone.

Ground Truth: Contradiction

This example was correctly classified by all prompting techniques. It represents a case where understanding the subtle distinction between ”external relations” in the premise and ”all executive powers” in the hypothesis is crucial. Even the Zero-Shot approach succeeds here, suggesting that some contradictions are readily apparent with minimal guidance.

The Agent-based approach provided the most comprehensive explanation:

Agent’s reasoning process (truncated)

SEMANTIC_PARSER analyzing premise
Key propositions: 1. The Republic of Ireland Act 1948 is an Act of the Oireachtas 2. This Act declared that the description of the Irish state was to be the Republic of Ireland 3. The Act vested in the President of Ireland the power to exercise the executive authority of the state in its external relations 4. This power is exercised on the advice of the Government of Ireland

ADVERSARIAL_DETECTOR
This example contains a quantifier shift. The premise states that the president has power over ”external relations” specifically, while the hypothesis claims ”all executive powers” were vested in the president. This is a classic adversarial technique where the scope is expanded from a limited domain to a universal claim.

This case demonstrates how even when all techniques reach the correct conclusion, the agent-based approaches provide more detailed and explicit reasoning, identifying specific adversarial patterns that could lead to misclassification.

5.3.2 Case 2: Agent Success, Others Fail

Example where only agent approaches succeed

Premise:
Lophophora williamsii is a small, spineless cactus with psychoactive alkaloids, particularly mescaline.

Hypothesis:
Lophophora williamsii is a blue plant.

Ground Truth: Neutral

This example was correctly classified as neutral by the Agent-based, RAG, and ReAct approaches, while Zero-Shot, Chain-of-Thought, and Generate Knowledge incorrectly labeled it as a contradiction. The key insight is recognizing that while cacti are typically green, the premise does not explicitly mention the plant's color, so the hypothesis's claim about it being blue is neither entailed nor contradicted.

The RAG approach's reasoning demonstrates why it succeeded:

RAG's reasoning (truncated)

The premise tells us that Lophophora williamsii is a cactus with certain properties (small, spineless, contains psychoactive alkaloids including mescaline).
The hypothesis claims that Lophophora williamsii is blue.
Looking at what we know from the premise, there is no information about the color of the cactus. While most cacti are green, the premise doesn't specify the color of this particular species. It's possible that this cactus could have a bluish tint or coloration under certain conditions, even though it's uncommon.
Since the premise doesn't give us information to confirm or deny the color, and the hypothesis is making a claim about a property not addressed in the premise, this is a neutral relationship.

This case highlights a key strength of agent-based approaches: they are more careful about distinguishing between common knowledge assumptions and explicit textual information. The retrieved knowledge in RAG and the structured reasoning in Agent-based and ReAct approaches helped avoid the common trap of assuming all cacti must be green.

5.3.3 Case 3: When All Techniques Fail

Example where all techniques fail

Premise:
Anniesland Court is a 24-storey residential tower block in the Anniesland district of Glasgow, Scotland. It was constructed in 1966 by George Wimpey for Glasgow City Council. The building's construction is of cast in-situ reinforced concrete, and it is the tallest tower building noted in the Pevsner's Guide to Glasgow. The building houses 140 tenants occupants and is located within a short distance to Anniesland rail station.

Hypothesis:
H. H. Holmes Partners designed Anniesland Court.

Ground Truth: Contradiction

This challenging example stumped all prompting techniques. While the premise mentions that George Wimpey constructed the building, it doesn't explicitly mention the designer. This creates ambiguity—could H.H. Holmes & Partners have designed it while Wimpey constructed it? All techniques either classified this as neutral or were uncertain.

The ReAct approach, despite its overall strong performance, reasoned as follows:

ReAct's reasoning (truncated)

Thought 3: Let's consider the relationship between the premise and hypothesis. Action 3: Compare the information about who was involved with Anniesland Court. Observation 3: The premise states that Anniesland Court ”was constructed in 1966 by George Wimpey for Glasgow City Council.” The hypothesis claims that ”H. H. Holmes Partners designed Anniesland Court.” The premise mentions who constructed the building (George Wimpey) but does not explicitly state who designed it. Construction and design are different aspects of building development.
Thought 4: Are there any potential traps or edge cases here? Action 4: Check if there's any information that would make these statements contradictory. Observation 4: The premise doesn't explicitly state who designed the building. While George Wimpey constructed it, they could have been working from designs created by H. H. Holmes Partners. Or the design could have been done by someone else entirely. Since the premise doesn't provide information about the designer, we can't determine if the hypothesis is true or false based solely on the given premise.
Final Thought: Since the premise doesn't mention who designed Anniesland Court, we cannot determine whether H. H. Holmes Partners designed it or not. Final Answer: neutral

This case highlights a limitation of current prompting approaches: they may struggle with inferences that require domain knowledge not explicitly mentioned in the text (in this case, understanding the typical relationship between construction companies and building design). It also reveals that even the most sophisticated prompting techniques cannot overcome certain limitations in the model's understanding when key information is implied rather than stated.

6 Conclusion

Our comprehensive study on enhancing performance on adversarial natural language inference through advanced prompting techniques offers several key insights based on our analysis of 100 balanced examples from the ANLI dataset:

1. **Progressive benefits of advanced prompting**: The substantial improvement from baseline Zero-Shot (54.0%) to more sophisticated techniques like Generate Knowledge (70.0%), Agent-based (76.0%), RAG (75.0%), and ReAct (80.0%) demonstrates that carefully designed prompts can significantly enhance model performance on challenging NLI tasks without requiring model retraining.
2. **Category-specific strengths**: Different prompting techniques show distinct strengths for particular inference categories. ReAct excels with entailment examples (85.3%), RAG performs best on neutral examples (78.8%), while the Agent-based approach demonstrates exceptional performance on contradiction detection (84.8%). This indicates that optimal prompting strategies should be tailored to the specific reasoning challenges of a given NLI task.
3. **Value of knowledge incorporation**: The Generate Knowledge and RAG approaches both demonstrate substantial improvements over simpler techniques by explicitly incorporating relevant knowledge about NLI principles and adversarial patterns. This suggests that enhancing prompts with domain-specific knowledge is a powerful strategy for improving reasoning performance.
4. **Agent vs. Non-Agent tradeoffs**: Our detailed comparison revealed significant advantages of agent-based approaches for complex reasoning tasks, particularly in reducing uncertain responses and handling challenging examples. However, these benefits come with computational costs (26.5% longer processing time), suggesting that the choice between agent and non-agent approaches should consider both performance requirements and resource constraints.
5. **Error pattern insights**: Our confusion matrix analysis reveals that more sophisticated prompting techniques not only improve accuracy but also reduce the frequency of ”un-

known" responses, from 32.0% with Zero-Shot to just 5.0% with ReAct. This indicates that structured prompting increases the model's confidence in making definitive classifications.

6. **Task complexity handling**: Agent-based approaches demonstrate more consistent performance across different levels of task complexity, maintaining a substantial advantage over traditional techniques for the most challenging examples. This suggests that the structured decomposition provided by agent prompting is particularly valuable for navigating complex reasoning challenges.

These findings highlight the importance of prompt engineering as a powerful and efficient approach for enhancing language model performance on complex reasoning tasks. By carefully designing prompts that activate relevant knowledge and guide models through structured analytical processes, we can substantially improve their ability to handle challenging adversarial examples without requiring additional training data or architectural changes.

Future work could explore combining prompting techniques with fine-tuning approaches, investigate adaptive prompting strategies that select different approaches based on input characteristics, or develop techniques to mitigate the specific error patterns identified in our analysis. Additionally, extending this comparative evaluation to other reasoning tasks beyond NLI would help identify which prompting techniques generalize well across different problem domains.

Acknowledgment

This is the second assignment for CSC6052/MDS5110/CSC5051/CSC4100, see details in https://nlp-course-cuhksz.github.io/.

References

[1] Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., & Kiela, D. (2020). Adversarial NLI: A New Benchmark for Natural Language Understanding. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4885–4901. Available at: https://arxiv.org/abs/1910.14599 and https://huggingface.co/datasets/facebook/anli

[2] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems 35*.

[3] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., K ü ttler, H., Lewis, M., Yih, W., Rocktschel, T., & others (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems 33*.

[4] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing Reasoning and Acting in Language Models. *arXiv preprint arXiv:2210.03629*.