LLMSR@XLLM25: Integrating Reasoning Prompt Strategies with Structural Prompt Formats for Enhanced Logical Inference

Le Duc Tai[1,2], Dang Van Thin[1,2],
[1]University of Information Technology-VNUHCM, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
23521374@gm.uit.edu.vn, thindv@uit.edu.vn

Abstract

This paper illustrates our team system approach in XLLM-ACL 2025 Task-III: LLM for Structural Reasoning (LLM-SR), aiming to solve both Task: Question parsing and CoT parsing. The process of extracting statements and evidence is similar to Discourse Parsing. Correct extraction of statements or evidence from the COT is crucial at the outset. Next, the pairwise relationship between a specific statement and its corresponding evidence is assessed (a statement should be followed by its related evidence from the CoT). Both semantic and lexical similarity are used to evaluate the accuracy of statements and evidence predictions. Finally, once a statement-evidence pair is correctly extracted, it is evaluated to determine whether the evidence can logically deduce the statement. To tackle Question Parsing and CoT Parsing, we implement and investigate various solutions, including (1) applying different structural prompt formats like JSON, Markdown, or XML. (2) utilising various prompt techniques: Few-shot, Chain of thought, and Multi-hop prompting. (3) Taking advantage of Natural Language Inference (NLI) model for the Statement Verification step. Our best official result is a 243.047 mean score for test phases A and B, and finally, we rank 7th on the final leaderboard.

1 Introduction

The advent of large language models (LLMs) has significantly advanced natural language processing, enabling sophisticated reasoning capabilities across diverse tasks. However, ensuring that these models produce structured, interpretable, and logically coherent reasoning remains a formidable challenge. Addressing this, the XLLM-ACL 2025 Task-III: LLM for Structural Reasoning (LLM-SR) focuses on evaluating the abilities of LLM to generate structured reasoning processes by parsing questions and corresponding chains of thought (CoT) into distinct components: major premises, minor premises, and their interrelations.

In this paper, we present the approach developed by our team, for the LLM-SR task. Our methodology targets two primary subtasks: **Question Parsing** and **CoT Parsing**. We conceptualise the extraction of statements and evidence as analogous to discourse parsing, emphasizing the accurate identification of these elements as a foundational step. Subsequently, we assess the pairwise relationships between specific statements and their corresponding evidence, ensuring that each statement is logically supported by its related evidence within the CoT.

To enhance the structural reasoning capabilities of LLMs, we investigate several strategies:

- **Structural Prompt Formats**: We explore the impact of different prompt formats, including JSON, Markdown, and XML, on the model's ability to parse and reason structurally.
- **Prompting Techniques**: We implement various prompting methods such as Few-shot learning, Chain of Thought (CoT), and Multi-hop prompting to guide the model's reasoning process.
- **Statement Verification**: We incorporate a Natural Language Inference (NLI) model to verify whether the extracted evidence logically entails the corresponding statements.

Our system achieved a mean score of 243.047 across test phases A and B, securing the 7th position on the final leaderboard. These results underscore the effectiveness of combining structured prompt formats with advanced prompting techniques and verification models to enhance the structural reasoning abilities of LLMs.

2 Related Work

Recent advancements in prompt engineering have demonstrated that providing large language models with a small number of in-context examples

Proceedings of the 1st Joint Workshop on Large Language Models and Structure Modeling (XLLM 2025), pages 310–316
August 1, 2025 ©2025 Association for Computational Linguistics

can yield strong zero and few-shot performance gains (Brown et al., 2020). Explicitly eliciting intermediate reasoning steps via Chain-of-Thought prompting has been shown to further boost complex arithmetic and commonsense reasoning (Wei et al., 2022). Techniques such as self-consistency decoding, which samples multiple reasoning paths and aggregates the most consistent answer, markedly improve Chain of Thought accuracy (Wang et al., 2022). Decomposition and multi hop strategies like Self Ask break down complex queries into subquestions for greater interpretability (Press et al., 2022). Hybrid reasoning–acting prompts (ReAct) interleave "Thought" and "Action" steps to ground LLMs in external environments or tools (Yao et al., 2022). Pipeline optimisations such as LM-BFF automate prompt template selection to refine few-shot tuning (Gao et al., 2020). Constraining LLM outputs to structured formats (JSON, XML) via schema-based or grammar-based decoding ensures machine-readable consistency for downstream extraction tasks (Lu et al., 2025). Methods from discourse parsing—segmenting text into elementary discourse units and labelling their rhetorical relations—provide algorithms analogous to statement and evidence extraction (Song and Liu, 2020). Finally, Natural Language Inference frameworks trained on large-scale corpora such as SNLI and MultiNLI underpin the verification of logical entailment between extracted evidence and hypothesis statements (Bowman et al., 2015; Williams et al., 2018).

3 Task Description

The XLLM-ACL 2025 Shared Task-III: LLM for Structural Reasoning (LLM-SR) challenges participants to generate a controllable and interpretable reasoning process via step-by-step inference (xll, 2025). It comprises two subtasks: *question parsing*, which extracts all conditions necessary for solving a given question, and *CoT parsing*, which segments a provided chain-of-thought into distinct statements and their corresponding evidence (xll, 2025). For each statement–evidence pair, systems must predict a binary verification label indicating whether the evidence logically entails the statement (xll, 2025). The training set contains 24 annotated examples derived from the LogiQA logical reasoning benchmark (Liu et al., 2020)(xll, 2025). Participants are restricted to using the Llama-3-8B-Instruct backbone model for all subtasks (xll, 2025)(Meta, 2024). Evaluation is performed on two public test phases (A and B), with Macro F1 computed over question parsing, statement parsing, statement–evidence pairing, and verification predictions (xll, 2025). Submissions are evaluated and scored through the Codabench platform, ensuring reproducibility and standardized scoring (Xu et al., 2022). The XLLM dataset (Shuyi-Zsy, n.d.) is conducted by the organizer.

4 Methodology

4.1 Prompt Format

Choosing an appropriate prompt format, such as JSON, Markdown, or XML, is fundamental to the design of reliable, interpretable, and maintainable LLM-based systems. The rigid key-value structure of JSON enforces unambiguous machine-readable output that simplifies downstream parsing and validation, but its verbosity can increase token usage and latency. Markdown, by contrast, offers a lightweight compromise: human-friendly readability paired with sufficient structural cues (headings, lists, code fences) that facilitate both developer inspection and automated post-processing. XML's tag-based hierarchy is well-suited for deeply nested or richly annotated content, allowing clear delineation of sections at the expense of larger prompt size and more complex parsing logic.

Systematic evaluation of these formats is necessary because prompt format choice can materially affect model behaviour, output consistency, and overall computational efficiency. Variations in format can influence the model's content planning heuristics, leading to differences in completeness, coherence, and error rates. Moreover, tokenization characteristics and schema overhead directly impact throughput and cost in production settings. By rigorously comparing JSON, Markdown, and XML prompts across tasks of varying complexity, researchers can quantify trade-offs between interpretability, performance stability, and resource consumption, thereby guiding the selection of the most appropriate format for a given application domain.

After experimenting, our team found out that leveraging the Markdown prompt format brought the best result on the question and CoT parsing task, as shown in the prompt sample below:

```
### Task:
You are an advanced reasoning
    assistant that extracts logical
    constraints, conditions, and
    final queries from complex
```

```
    reasoning questions. Given a
    question, you will return a
    structured list of parsed
    conditions.

### **Instructions:**
1. Identify the **problem context**
    (e.g., participants, locations,
    objects).
2. Extract **explicit logical
    conditions** (marked by "if", "
    then", "must", "cannot", "
    different", not,etc.).
3. Focus on **bullet point numbers**
     because logical constraint
    often start with **bullet point
    numbers**.
4. Identify the **final question
    statement** that requires
    solving.
5. Parse output MUST only contain
    information in provided question
     and do not hallucinate.
6. **Check if the final query
    contains logical constraints**
    and extract them separately.
7. Format your response as a
    structured JSON output.
```

4.2 Few-shot Prompting

Few-shot prompting leverages a small set of annotated examples to guide the model toward the desired output structure and reasoning patterns without extensive fine-tuning. By providing representative question analysis pairs, the model learns to generalize the extraction of premises, evidence, and their logical relations directly from the prompt context. This approach is highly efficient: it requires minimal manual annotation effort compared to fully supervised training, and it can be adapted to new subtasks or domains by swapping in a few new exemplars.

In the context of LLM-SR, few-shot prompting enhances both accuracy and reliability. Exemplars that demonstrate correct statement identification and evidence pairing serve as implicit templates, reducing ambiguity in model predictions and improving consistency across instances. Moreover, few-shot formats naturally encourage the model to attend to relevant structural cues, such as delineated premises or marked evidence segments, thus aligning its internal content planning with the requirements of question parsing and CoT parsing. From a computational standpoint, the overhead of including a handful of examples in the prompt is marginal relative to the gains in output precision, making few-shot prompting a cost-effective technique for rapid prototyping and iterative system development.

As we attempted, we observed that there are types of questions in the dataset, and we listed out those types, then provided example as few-shot prompting for the prompt. As a result, we saw improvement in F1-score when evaluating, but since providing too many examples, the prompt sometimes returns biased output, which is too stuck with the example that we set in the prompt. To tackle this problem, we build a component that check whether the output was too different from the given question or CoT, then it must re-generate another output until it meets the similarity threshold with the given question or CoT.

4.3 Chain of Thought

Chain-of-Thought prompting guides the model to generate intermediate reasoning steps explicitly, thereby transforming an opaque prediction process into a multi-step, interpretable inference chain. By eliciting rationales before producing final outputs, CoT prompts align the model's internal content planning with the structural requirements of the LLM-SR task, facilitating accurate identification of premises, supporting evidence, and their logical relationships. This explicit decomposition of reasoning not only improves the model's attention to critical details such as the dependencies between question conditions and derived conclusions but also enables straightforward error analysis and targeted prompt refinement.

From an efficiency standpoint, CoT prompting leverages the pre-trained reasoning abilities of LLMs without additional fine-tuning, requiring only the inclusion of a few illustrative CoT exemplars in the prompt. The marginal increase in prompt length is outweighed by gains in accuracy and consistency, particularly for complex multi-step inferences inherent to structural reasoning. Moreover, the generated chains of thought can be post-processed to automatically extract statements and evidence segments, thereby streamlining the end-to-end pipeline for question parsing and CoT parsing. Consequently, Chain-of-Thought prompting represents a cost-effective and scalable technique for enhancing both the interpretability and performance of LLM-based structural reasoning systems.

Even though Chain of thought prompting has many effective aspects, in our approach, we only use Chain of thought prompting for generate knowledge,e which will be fed into the next prompt for

multi-hop prompting technique.

4.4 Multi-hop Prompting

Multi-hop prompting decomposes complex reasoning tasks into a sequence of dependent subquestions, each answered in turn to build a coherent inference chain. This structured decomposition aligns naturally with the LLM-SR subtasks of question parsing and CoT parsing, as it forces the model to identify intermediate premises and evidence at each hop. By guiding the model to focus on one inference step at a time, multi-hop prompts reduce hallucinations and improve the precision of statement–evidence pairing. Moreover, the modular nature of multi-hop prompting enables flexible adaptation: new sub-questions can be added or refined without retraining, and individual hops can be optimized for efficiency, making it a cost-effective strategy for scalable structural reasoning.

Multi-hop prompting breaks down complex queries into sequential sub-questions, guiding the model to iteratively extract statements and their corresponding evidence. By isolating each inference step, this technique improves the accuracy of statement–evidence alignment, minimizes spurious connections, and enables targeted refinement of individual hops without retraining, making it an efficient strategy for our LLM-SR task.

4.5 Natural Language Inference (NLI)

Natural Language Inference (NLI) provides an effective and efficient mechanism for verifying whether an extracted evidence segment logically entails its paired statement. By framing verification as an entailment classification task, we leverage pre-trained NLI models to score statement–evidence pairs without additional fine-tuning, minimizing annotation overhead and development time. The binary entailment output directly aligns with the LLM-SR verification requirement, enabling fast, consistent judgments and straightforward integration into the parsing pipeline. Moreover, NLI models exhibit strong generalization across domains, ensuring robust performance even when evidence and statement formulations vary in wording or structure. This approach streamlines the verification step and enhances overall system reliability with minimal computational and engineering cost.

5 Full Pipeline

This is our full best pipeline, which is shown in 1.

The proposed pipeline constitutes a modular and interpretable architecture tailored for the LLM-SR task, effectively addressing both question parsing and statement–evidence verification through a sequence of structured components. It begins with an input consisting of a question and its corresponding Chain-of-Thought (CoT) rationale. A parsing prompt is applied to extract candidate statements and evidence segments from the CoT, yielding a structured intermediate representation. This initial decomposition step is critical, as it transforms unstructured natural language into discrete units that downstream components can process more reliably.

Next, a similarity check module evaluates the lexical and semantic coherence between extracted statements and their corresponding evidence spans. This filtering mechanism ensures that only aligned pairs proceed to the next stage, thereby minimizing noise and reducing the likelihood of spurious relations. Following this, the pipeline incorporates two reasoning pathways in parallel: Chain-of-Thought prompting and Multi-hop prompting. Chain-of-Thought prompting improves interpretability and promotes stepwise deduction by explicitly modeling intermediate reasoning steps. In contrast, Multi-hop prompting decomposes complex inference into smaller, interdependent sub-questions, enabling more accurate retrieval of distributed evidence and enhancing logical consistency.

The outputs from these reasoning modules are routed into a Natural Language Inference (NLI) model, which performs the verification step by determining whether each evidence segment entails its associated statement. This dedicated verification layer isolates the decision-making process from generation, improving both reliability and transparency. By leveraging pre-trained NLI models, the system achieves strong verification performance without additional supervision.

Overall, this pipeline exemplifies best practices in prompt-based LLM system design. Its modular structure allows for independent tuning and component replacement, fostering adaptability and ease of maintenance. The integration of structured prompting strategies, semantic similarity filtering, and NLI-based verification results in a robust and scalable solution for structural reasoning tasks. Moreover, the pipeline supports transparency and interpretability at each stage, making it suitable for high-stakes domains where explanation and traceability are essential.

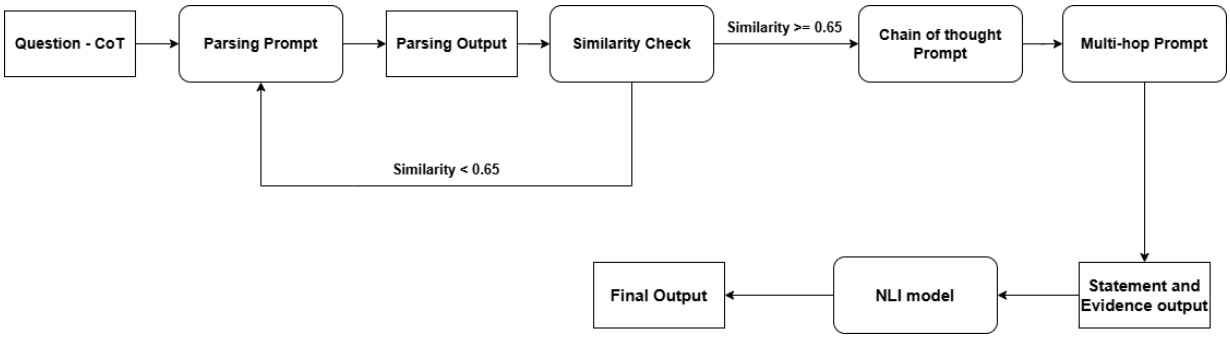

Figure 1: Full pipeline of LLM

5.1 Experimental Settings

Our experimental framework is designed to evaluate structural reasoning capabilities across multiple prompting strategies for the LLM-SR task. We utilize the public test set from the XLLM-ACL 2025 Task-III dataset, formatted in JSON and parsed using a custom data loader. The system is built around Meta-Llama-3-8B-Instruct, a state-of-the-art causal language model, accessed via HuggingFace's Transformers library. The model is loaded using 4-bit quantization with mixed-precision inference to optimize computational efficiency while preserving performance.

To facilitate robust text generation and parsing, we use a Transformer-based pipeline configured for causal language modeling. Chain-of-Thought (CoT), Few-shot, and Multi-hop prompting strategies are incorporated into distinct parsing modules to support statement extraction, evidence retrieval, and verification. Each prompt format (e.g., JSON, Markdown, XML) is evaluated across the different reasoning tasks to assess structural sensitivity and effectiveness.

6 Main Results

Table 1 summarises the performance of seven prompt configurations on the structural reasoning subtasks. Several clear patterns emerge:

First, the combination of Few-shot, Chain-of-Thought, Multi-hop prompting, NLI verification, and similarity checking, all delivered in a Markdown format with four prompt calls, achieves the highest overall accuracy (Question Parsing: 67.96%, Statement Parsing: 39.21%, Statement–Evidence Matching: 12.04%, Reasoning: 3.83). This demonstrates that layering multiple complementary prompting techniques yields significant gains, particularly for the most challenging subtasks of identifying and aligning statements with their evidential support.

Second, reducing the prompt format to pure Markdown without the similarity check module causes a modest drop in performance (Question Parsing: 64.75%, Statement Parsing: 29.90%, Statement–Evidence Matching: 9.40%, Reasoning: 2.63), illustrating the value of the auxiliary filtering stage. The JSON format, when used with the full suite of prompting techniques, further decreases Question Parsing accuracy to 61.40%, while slightly improving Statement Parsing (30.36%) but reducing Statement–Evidence matching (8.70%) and overall reasoning quality (3.20). This suggests that Markdown's human-readable cues better guide the model's content planning compared to the more rigid JSON schema.

Third, limiting the system to only Few-shot prompting with NLI (two prompt calls) yields mixed results: Markdown achieves stronger Question Parsing (64.15%) but lower Statement–Evidence alignment (1.85%) and minimal reasoning depth (0.90), whereas JSON boosts Statement–Evidence Matching (11.70%) at the expense of Question Parsing (59.28%) and reasoning quality (4.50). This indicates that NLI verification alone can compensate for reduced prompting complexity in pairing statements and evidence, but at a cost to holistic parsing performance.

Finally, the simplest Few-shot–only configurations (two or four calls) produce the lowest scores across all subtasks (Question Parsing: 56.02–59.24%, Statement Parsing: 29.17–38.50%, Statement–Evidence Matching: 2.07–2.29%, Reasoning: 1.07–1.84), confirming that advanced prompting strategies are critical to unlock the full structural reasoning capabilities of LLMs.

In summary, these results underscore the effectiveness of integrating multiple prompting techniques within a Markdown format and highlight

the trade-offs inherent in prompt format selection, prompt complexity, and verification strategy. The highest-performing configuration (four calls, Markdown, full-technique suite) is the reference point for further improvements in structural reasoning pipelines. Finally, our best score on test set B are 73.24, 47.07, 15.59, and 10.22. Our team achieve rank 7th i XLLM-ACL 2025 Task-III: LLM for Structural Reasoning (LLM-SR).

7 Conclusion and Future Work

In this study, we presented a comprehensive and modular pipeline for addressing the LLM-SR task, targeting both question parsing and Chain-of-Thought (CoT) parsing subtasks. Through a series of experiments, we demonstrated that the integration of structured prompt formats, advanced prompting strategies (Few-shot, Chain-of-Thought, Multi-hop), and a dedicated NLI-based verification step significantly enhances the model's structural reasoning performance. Our analysis highlights the effectiveness of Markdown as a prompt format and the importance of leveraging multiple complementary reasoning techniques to ensure accurate statement–evidence alignment and logical verification.

In future work, we plan to explore dynamic prompt selection mechanisms that adapt based on question complexity, as well as integrating retrieval-augmented generation (RAG) components to support knowledge-intensive reasoning. We are also interested in fine-tuning or instruction-tuning smaller models to perform verification and parsing steps more efficiently.

Acknowledgements

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

References

2025. Xllm-acl 2025 shared task-iii: Llm for structural reasoning (llm-sr). https://xllms.github.io/LLMSR/#.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and *et al.* 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3816–3830.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *CoRR*, abs/2007.08124.

Yaxi Lu, Haolun Li, Xin Cong, Zhong Zhang, Yesai Wu, Yankai Lin, Zhiyuan Liu, Fangming Liu, and Maosong Sun. 2025. Learning to generate structured output with schema reinforcement learning. *Preprint*, arXiv:2502.18878.

Meta. 2024. Meta llama 3.0 8b instruct. https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *Preprint*, arXiv:2210.03350.

Shuyi-Zsy. n.d. Llmsr dataset. Accessed: 2025-05-15.

Wei Song and Lifeng Zhu Liu. 2020. Representation learning in discourse parsing: A survey. *Science China Technological Sciences*, 63(10):1921–1946.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Zhen Xu, Sergio Escalera, Isabelle Guyon, Adrien Pavão, Magali Richard, Wei-Wei Tu, Quanming Yao, and Huan Zhao. 2022. codabench: flexible, easy-to-use and reproducible benchmarking platform. *arXiv preprint arXiv:2110.05802*.

Shunyu Yao, Dian Zhao, Jeffrey Yu, Izhak Shafran, Karthik Narasimhan, and Yue Cao. 2022. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Prompt Calls	4
Prompt Format	Markdown
Techniques	Few-shot, Chain-of-Thought, Multi-hop, NLI, Similarity check
Scores	Question Parsing: 67.96, Statement Parsing: 39.21, Statement–Evidence: 12.04, Reasoning: 3.83
Prompt Calls	4
Prompt Format	Markdown
Techniques	Few-shot, Chain-of-Thought, Multi-hop, NLI
Scores	Question Parsing: 64.75, Statement Parsing: 29.90, Statement–Evidence: 9.40, Reasoning: 2.63
Prompt Calls	4
Prompt Format	JSON
Techniques	Few-shot, Chain-of-Thought, Multi-hop, NLI
Scores	Question Parsing: 61.40, Statement Parsing: 30.36, Statement–Evidence: 8.70, Reasoning: 3.20
Prompt Calls	2
Prompt Format	Markdown
Techniques	Few-shot, NLI
Scores	Question Parsing: 64.15, Statement Parsing: 25.30, Statement–Evidence: 1.85, Reasoning: 0.90
Prompt Calls	2
Prompt Format	JSON
Techniques	Few-shot, NLI
Scores	Question Parsing: 59.28, Statement Parsing: 39.21, Statement–Evidence: 11.70, Reasoning: 4.50
Prompt Calls	2
Prompt Format	Markdown
Techniques	Few-shot
Scores	Question Parsing: 56.02, Statement Parsing: 38.50, Statement–Evidence: 2.07, Reasoning: 1.07
Prompt Calls	4
Prompt Format	Markdown
Techniques	Few-shot
Scores	Question Parsing: 59.24, Statement Parsing: 29.17, Statement–Evidence: 2.29, Reasoning: 1.84

Table 1: Results on Test set A in different approaches