

LLMSR@XLLM25: An Empirical Study of LLM for Structural Reasoning

Xinye Li[†] Mingqi Wan[†] Dianbo Sui^{*}

Harbin Institute of Technology

{lixinye, 2022211876}@stu.hit.edu.cn, suidianbo@hit.edu.cn

Abstract

We present Team *asdfs123*’s submission to the LLMSR@XLLM25 shared task, which evaluates large language models on producing fine-grained, controllable, and interpretable reasoning processes. Systems must extract all problem conditions, decompose a chain of thought into statement–evidence pairs, and verify the logical validity of each pair. Leveraging only the off-the-shelf **Meta-Llama-3-8B-Instruct**, we craft a concise few-shots, multi-turn prompt that first enumerates all conditions and then guides the model to label, cite, and adjudicate every reasoning step. A lightweight post-processor based on regular expressions normalises spans and enforces the official JSON schema. Without fine-tuning, external retrieval, or ensembling, our method ranks **5th** overall, achieving macro-F₁ scores on par with substantially more complex and resource-consuming pipelines. We conclude by analysing the strengths and limitations of our approach and outlining directions for future research in structural reasoning with LLMs. Our code is available at <https://github.com/asdfs123/LLMSR-asdfs123>.

1 Introduction

Large language models (LLMs) have recently shown impressive performance on complex reasoning tasks, spurred in part by *Chain-of-Thought* (CoT) prompting, which asks the model to externalise intermediate steps before giving an answer (Wei et al., 2023). Subsequent variants—such as zero-shot CoT (Kojima et al., 2022), self-consistency decoding (Wang et al., 2023), tree-of-thought search (Yao et al., 2023), and automatically generated demonstrations (Zhang et al., 2022)—further boost accuracy, yet these free-form rationales remain difficult to evaluate and prone to hallucinations (Akbar et al., 2024).

The LLMSR@XLLM25 shared task tackles this limitation by framing reasoning as a constrained CoT process: systems must (i) extract every explicit problem condition, (ii) segment a rationale into aligned **statement–evidence** pairs, and (iii) judge whether each evidence span **logically entails** its statement. Such fine-grained structure “improves the transparency and reliability of the process” (task description) and enables detailed diagnosis of model behaviour. Moreover, the step-level labels provide dense supervision for Process Reward Modeling (PRM), which optimises *how* a solution is reached rather than merely *what* answer is produced (Uesato et al., 2022; Lightman et al., 2023).

Structured parsing of reasoning brings three concrete benefits. First, it enhances **debuggability**: developers can pinpoint the exact step where a hallucination or logical slip occurs. Second, it supplies explicit training signals for PRM, shown to yield more coherent and truthful solutions on mathematical benchmarks (Lightman et al., 2023). Third, it promotes **trustworthy AI**: users can audit or amend individual steps, a requirement for safety-critical deployments and formal logic tasks such as EntailmentBank proofs (Dalvi et al., 2021) or LogicBench diagnostics (Parmar et al., 2024).

In this report we present Team *asdfs123*’s lightweight submission, which relies solely on the untuned **Meta-Llama-3-8B-Instruct** (Meta AI, 2024). A compact few-shot, multi-turn prompt guides the model through all three subtasks, while a minimal post-processor enforces the official JSON schema. Despite its simplicity, our approach ranks **5th** overall, demonstrating that careful prompt design and constrained reasoning can rival far more elaborate pipelines.

^{*}Dianbo Sui is the corresponding author.

[†]Equal contribution.

2 Related Work

2.1 Chain-of-Thought Prompting

Chain-of-Thought (CoT) prompting has emerged as a powerful method to enhance multi-reasoning in large language models (LLMs). Initial studies showed significant improvements by simply adding "Let's think step by step" to zero-shot prompts (Kojima et al., 2022). Self-consistency further boosts robustness by generating multiple reasoning chains and selecting the most consistent response (Wang et al., 2023). Least-to-Most prompting addresses complex problems by decomposing them into simpler subproblems, achieving near-perfect accuracy on challenging tasks (Zhou et al., 2023).

However, CoT prompting can produce logically flawed reasoning steps, reaching correct answers through invalid logic (Zelikman et al., 2022; Golovneva et al., 2023). The Tree of Thoughts framework mitigates this by organizing reasoning into a search tree, allowing systematic backtracking and evaluation of alternative reasoning paths (Yao et al., 2023). Incorporating knowledge-graph-based verification also improves reliability (He et al., 2025; Jiang et al., 2023).

Recent benchmarks focus on evaluating CoT quality beyond answer accuracy, using validity and redundancy metrics to assess reasoning step-by-step (Xia et al., 2025; Chen et al., 2025). These approaches emphasize the need for tighter integration between reasoning generation and verification.

2.2 Parsing

Turning natural language into structured representations is a prerequisite for dependable reasoning. ProgPrompt steers LLMs to emit code-like blocks of comments, actions, and assertions for situated robot planning (Singh et al., 2022). Self-Ask improves interpretability by decomposing a complex query into solvable sub-questions and then composing their answers (Press et al., 2023). Coupling LLMs with Answer Set Programming lets a logic engine verify every inferred rule, boosting robustness (Yang et al., 2023). RaLU aligns CoT spans with formal logic units and checks them via external solvers (Li et al., 2025).

For discourse-level parsing, Rhetorical Structure Theory (RST) models text coherence via nucleus-satellite relations (MANN and THOMPSON, 1988). Early algorithms split texts into Elementary Discourse Units and attached rhetorical relations—sometimes without explicit markers (Marcu,

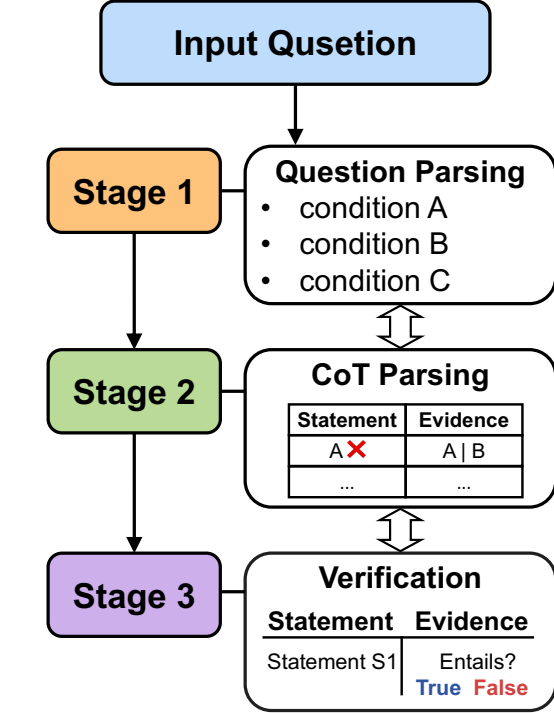


Figure 1: Illustration of the three-stage LLM-SR Task. (In our implementation, Verification is executed within the CP stage.)

1998). Enhanced RST (eRST) extends this to graphs with non-projective, concurrent relations and both implicit and explicit signals, offering more flexible, explainable structures (Zeldes et al., 2024).

2.3 Process Reward Model

Previous studies have demonstrated that process supervision maintains reasoning consistency better than outcome supervision, and conceptualized Process Reward Models (PRMs) to reduce logical errors (Uesato et al., 2022; Lightman et al., 2023). To mitigate the cost of manual annotations, recent approaches automatically retrieve similar solution steps to generate fine-grained, step-level labels—facilitating both verification and PPO-based reinforcement learning without human supervision (Wang et al., 2024).

Building on the foundational PRM framework, several works have further advanced process reward modeling. Tree-based preference learning constructs reasoning trees via best-first search and trains verifiers using paired step-level preferences (He et al., 2024). More recently, CFPRM (Hu et al., 2025) introduces a coarse-to-fine strategy that first merges adjacent steps into coarse-grained win-

dows and then refines them into fine-grained units. This hierarchical method mitigates redundancy in LLM-generated reasoning while enabling training across multiple levels of granularity.

3 Methodology

3.1 Pipeline Overview

Our system follows the three-stage workflow mandated by the LLM-SR task (Figure 1):

1. **Question Parsing (QP).** The model enumerates every explicit condition of the problem as an ordered list.
2. **CoT Parsing & Verification (CP).** Given the question, its Chain-of-Thought (CoT) rationale, and the QP output, the model simultaneously (i) segments the rationale into **statement-evidence pairs** and (ii) judges whether each evidence span **logically entails** its statement.

All stages run on the untuned original **Meta-Llama-3-8B-Instruct**. Instead of parameter fine-tuning we rely on *few-shot in-context learning* (ICL) with a multi-turn dialogue template (§3.2). A deterministic post-processor (§3.4) validates and cleans the raw generations, after which we complete the full public test set in under ten minutes.

3.2 Prompt Engineering

Few-shot demonstrations. We hand-pick two QP and three CP exemplars that jointly cover most patterns. During inference these demonstrations precede the test instance verbatim.

Three-turn template. Each call is cast as a short conversation:

1. **SYSTEM:** global rules, including format restrictions.
2. **USER:** the problem text plus the explicit request (QP or CP).
3. **ASSISTANT:** the model’s structured JSON answer.

Because CP depends on the extracted conditions, we invoke the model twice per instance: first for QP, and then for a single CP call which jointly performs CoT parsing and the verification step, with the QP list appended to the user prompt.

3.3 Robust JSON Output

Llama-3 occasionally produces ill-formed JSON—extra quotes, missing commas, or unclosed braces—which crashes the official scorer. By enclosing every demonstration answer in a fenced ````json . . . ```` block and explicitly instructing the model to output valid JSON only, we cut the unparseable rate on the dev set from 16 % to just 2 %. The few residual errors are corrected or flagged by our post-processor.

3.4 Post-processing

A lightweight Python script performs:

1. **Schema check:** every object must contain statement, evidence, and boolean verification.
2. **Normalisation:** trim bullets, stray whitespace, smart quotes, trailing punctuation; merge duplicate conditions.
3. **Alignment:** if $\#statements \neq \#evidence$, align by order; otherwise flag (none observed on dev/test).

3.5 Efficiency Rationale

The task rewards both answer correctness and reasoning quality. We show that careful prompt design plus minimal hygiene techniques already yields a top-5 macro- F_1 without external retrieval or fine-tuning, providing a strong, reproducible baseline for future work on PRM.

4 Experiments

We conduct all experiments on the official LLMsr@XLLM25 test sets¹. The shared task provides a *fine-grained* Chain-of-Thought (CoT) analysis corpus derived from LogiQA (Liu et al., 2021). It contains only 24 fully annotated training instances, each accompanied by both *question-parsing* and *CoT-parsing* labels. From the 24 training instances, we heuristically select a small subset of demonstrations that spans the major logical patterns; these serve as the few-shot exemplars in our prompts.

The evaluation follows a two-stage protocol. First, we perform a k -shot ablation for **Question Parsing** (QP), varying the number of in-context demonstrations. After selecting the best QP setting,

¹<https://huggingface.co/datasets/shuyi-zsy/LLMSR/tree/main/llmsr>

we keep it fixed and sweep k again for **CoT Parsing & Verification** (CP) to determine its optimal demonstration budget.

4.1 Phase 1: Selecting the Question-Parsing Shot Count

Table 1 shows QP results with $k \in \{1, 2, 3, 4\}$. Macro-F₁ peaks at **0.7526** with **2-shot**. Adding a third or fourth example degrades performance, presumably because the longer prompt dilutes salient patterns and pushes relevant context tokens farther from the model’s attention window.

| Shots (k) | Question_Macro_F1 |
|---------------|-------------------|
| 1 | 0.6707 |
| 2 | 0.7526 |
| 3 | 0.7281 |
| 4 | 0.7061 |

Table 1: Few-shot ablation for Question Parsing.

Given its clear advantage, we *fix* $k=2$ for all subsequent QP calls. The extracted condition list is then passed as additional context to the CP stage.

4.2 Phase 2: Tuning CoT Parsing & Verification

After fixing the QP stage at two demonstrations, we sweep the shot count for CoT Parsing. Table 2 shows that **3-shot** strikes the best trade-off, yielding the highest *Statement_Macro_F1* as well as the strongest pair-level and reasoning scores. Adding a fourth example brings only marginal gains and in some cases degrades performance, presumably because the longer prompt pushes relevant tokens farther from the model’s attention window.

| CP Shots | Stmt _{F1} | Stmt+Ev _{F1} | Reasoning _{F1} |
|----------|--------------------|-----------------------|-------------------------|
| 1 | 0.3066 | 0.0726 | 0.0391 |
| 2 | 0.1816 | 0.0860 | 0.0250 |
| 3 | 0.3304 | 0.1385 | 0.0782 |
| 4 | 0.2978 | 0.0976 | 0.0518 |

Table 2: CoT Parsing & Verification with 2-shot QP fixed. “Stmt” = Statement_Macro_F1, “Stmt+Ev” = Statement_Evidence_Macro_F1.

4.3 Final Configuration

The combination of **2-shot QP** and **3-shot CP** constitutes our submission. This hybrid setup achieves

the highest overall macro-F₁ on the public leaderboard while preserving the system’s lightweight. The results highlight two insights: (1) QP and CP favour different demonstration budgets, and (2) carefully tuning each stage separately beats a single fixed prompt size.

We report our final experimental results in Table 3, which include the Test A and Test B phase scores on the official LLM@XLLM25 test sets.

| Phase | Question _{F1} | Stmt _{F1} | Stmt+Ev _{F1} | Reasoning _{F1} |
|--------|------------------------|--------------------|-----------------------|-------------------------|
| Test A | 75.26 | 33.04 | 13.85 | 7.82 |
| Test B | 75.33 | 47.26 | 20.17 | 11.64 |

Table 3: Macro-F1 scores on four evaluation criteria for Test A and Test B phases. “Stmt” = Statement_Macro_F1, “Stmt+Ev” = Statement_Evidence_Macro_F1.

5 Discussion

5.1 Key Insights from the Shared Task

The LLM@XLLM25 shared task offers a concrete sandbox for **controllable** and **transparent** reasoning. By forcing systems to expose every condition, align each statement with explicit evidence, and render a step-level entailment verdict, the task goes well beyond conventional answer-only evaluation. Our experiments confirm three central insights:

1. **Structural reasoning is promising yet non-trivial.** Even an untuned 8B model can reliably parse conditions (§4, Phase 1), but struggles to decompose and verify chains of thought.
2. **Larger does not (yet) mean satisfactory.** Informal leaderboard comparisons indicate that more elaborate, resource-heavy pipelines still fall short. The bottleneck is not extraction but *logical adjudication*.

5.2 Limitations of Llama-3-8B

Meta-Llama-3-8B scores well on QP but falters on logic: it hallucinates evidence, merely paraphrases conditions, and mishandles negation, dragging down Statement–Evidence and Reasoning F₁. These errors persist despite prompt tuning and JSON guards, implying the bottleneck lies in the model’s logic rather than the interface.

5.3 Future Work

Stronger verifiers. Verification may need a more capable judge (e.g., GPT-4o, Claude 3) detached from the generator.

Lightweight entailment modules. Training a small, dedicated critic on synthetic entailment pairs—à la CoT-Critic—could boost step-level faithfulness.

Process Reward Models (PRMs). The extracted structures are ideal supervisory signals for PRMs. Iteratively refining the generator with PRM feedback may tighten the link between evidence and statements, increasing coherence without brute-force scaling.

5.4 Takeaway

The shared task shows that structured reasoning is a feasible yet unsolved frontier for LLMs. Our minimal system serves as a proof of concept; progress now hinges on developing (i) stronger or specialised verifiers and (ii) learning paradigms that reward *how* a conclusion is reached, not merely *what* it is. We believe these directions will be pivotal for deploying LLMs in settings where transparency and trustworthiness are non-negotiable.

6 Conclusion

We showed that a carefully crafted, few-shot prompting pipeline—backed by lightweight post-processing—can tackle the LLMSR@XLLM25 shared task without fine-tuning or external tools, ranking 5th overall. While Meta-Llama-3-8B handles condition extraction well, its verification accuracy remains limited, underscoring the need for stronger or specialised reasoners and process-level training signals. Future work should pair stronger base models with dedicated entailment critics and reward models that explicitly value step-by-step correctness.

References

- Shayan Ali Akbar, Md Mosharaf Hossain, Tess Wood, Si-Chi Chin, Erica M Salinas, Victor Alvarez, and Erwin Cornejo. 2024. [HalluMeasure: Fine-grained hallucination measurement using chain-of-thought reasoning](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15020–15037, Miami, Florida, USA. Association for Computational Linguistics.
- Guizhen Chen, Weiwen Xu, Hao Zhang, Hou Pong Chan, Chaoqun Liu, Lidong Bing, Deli Zhao, Anh Tuan Luu, and Yu Rong. 2025. Finereason: Evaluating and improving llms’ deliberate reasoning through reflective puzzle solving. arXiv preprint arXiv:2502.20238.
- Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. [Explaining answers with entailment trees](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7358–7370, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. [Roscoe: A suite of metrics for scoring step-by-step reasoning](#). Preprint, arXiv:2212.07919.
- Jiashu He, Mingyu Derek Ma, Jinxuan Fan, Dan Roth, Wei Wang, and Alejandro Ribeiro. 2025. Give: Structured reasoning of large language models with knowledge graph inspired veracity extrapolation. arXiv preprint arXiv:2410.08475.
- Mingqian He, Yongliang Shen, Wenqi Zhang, Zeqi Tan, and Weiming Lu. 2024. [Advancing process verification for large language models via tree-based preference learning](#). Preprint, arXiv:2407.00390.
- Yulan Hu, Ge Chen, Jinman Zhao, Sheng Ouyang, and Yong Liu. 2025. [Coarse-to-fine process reward modeling for mathematical reasoning](#). Preprint, arXiv:2501.13622.
- Jinhao Jiang, Kun Zhou, Xin Zhao, Yaliang Li, and Jirong Wen. 2023. Reasoninglm: Enabling structural subgraph reasoning in pre-trained language models for question answering over knowledge graph. In *EMNLP*, pages 3721–3735.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS ’22*, Red Hook, NY, USA. Curran Associates Inc.
- Cheryl Li, Tianyuan Xu, and Yiwen Guo. 2025. Reasoning-as-logic-units: Scaling test-time reasoning in large language models through logic unit alignment. arXiv preprint arXiv:2502.07803.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. [Let’s verify step by step](#). Preprint, arXiv:2305.20050.
- Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2021. Logiqa: a challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI’20*.
- WILLIAM C. MANN and SANDRA A. THOMPSON. 1988. [Rhetorical structure theory: Toward a functional theory of text organization](#). *Text - Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

- Daniel C. Marcu. 1998. *The rhetorical parsing, summarization, and generation of natural language texts*. Ph.D. thesis, CAN. AAINQ35238.
- Meta AI. 2024. [Meta-llama-3-8b-instruct model card](#).
- Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. 2024. [LogicBench: Towards systematic evaluation of logical reasoning ability of large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13679–13707, Bangkok, Thailand. Association for Computational Linguistics.
- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of EMNLP*, pages 5687–5711.
- Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. 2022. Progprompt: Generating situated robot task plans using large language models. *arXiv preprint arXiv:2209.11302*.
- Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. [Solving math word problems with process- and outcome-based feedback](#). *Preprint*, *arXiv:2211.14275*.
- Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Su. 2024. [Math-shepherd: Verify and reinforce llms step-by-step without human annotations](#). *Preprint*, *arXiv:2312.08935*.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [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 Chi, Quoc Le, and Denny Zhou. 2023. [Chain-of-thought prompting elicits reasoning in large language models](#). *Preprint*, *arXiv:2201.11903*.
- Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. 2025. Evaluating mathematical reasoning beyond accuracy. *arXiv preprint arXiv:2404.05692*.
- Zhun Yang, Adam Ishay, and Joohyung Lee. 2023. Coupling large language models with logic programming for robust and general reasoning from text. *arXiv preprint arXiv:2307.07696*.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. [Tree of thoughts: Deliberate problem solving with large language models](#). *Preprint*, *arXiv:2305.10601*.
- Amir Zeldes, Tatsuya Aoyama, Yang Janet Liu, Siyao Peng, Debopam Das, and Luke Gessler. 2024. [erst: A signaled graph theory of discourse relations and organization](#). *Preprint*, *arXiv:2403.13560*.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with self-consistency. In *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488.
- Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. [Automatic chain of thought prompting in large language models](#). *Preprint*, *arXiv:2210.03493*.
- Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. [Least-to-most prompting enables complex reasoning in large language models](#). *Preprint*, *arXiv:2205.10625*.