Evaluating Prompting Architectures for Linguistic Reasoning in LLMs: Based on Linguistic Olympiad Problems

Gyeongseo Hwang
Department of Linguistics
2023-10775
rudtj0801@snu.ac.kr

Abstract

This paper investigates whether large language models (LLMs) can truly infer symbolic linguistic rules or merely exploit statistical patterns, by comparing three prompting architectures—Tree-of-Thoughts, iterative updating method, and non-iterative method—on PuzzLing dataset. This paper introduces two distilled knowledge elements, corpus (low-resource language's word-English interpretation mappings) and explicit inference rules, and evaluates their effects. The experiments show that the ToT variant requires more computation but fails to surpass simpler designs. The non-iterative architecture that retains both corpus and rules exceeds the performance of the iterative approach, indicating that repeated LLM self-revision yields minimal gains. To add, at least in low-performance models like GPT-3.5-turbo, corpus extraction provides greater accuracy improvements than rule extraction, showing the importance of lexical grounding over formal rule lists in LLMs' linguistic reasoning.
Moreover, this paper shows synergy between raw example sentences and distilled knowledge. Models achieve peak accuracy when both 'example sentences' and 'corpus + rules' are provided in the prompt. These findings suggest that effective in-context learning for linguistic rule induction hinges on balancing concrete lexical examples with lightweight pattern extraction, while avoiding costly abstract reasoning loops.

1 Introduction

The central question of this paper is whether LLMs are merely probabilistic machines based on statistical patterns, or whether they can truly infer linguistic rules. Traditional linguistic theories, particularly generative grammar, have characterized language as a system composed of symbolic elements and explicit rules. In contrast, LLMs operate without explicit rules, learning context-dependent probability distributions through the optimization of a vast number of parameters.

This raises the question of whether such models can effectively capture the rule-governed nature of human language—particularly its compositionality and capacity for generalization. Understanding the generalization capabilities and limitations of LLMs not only helps to illuminate the internal mechanisms of these black-box models, but also offers a valuable theoretical framework for exploring the cognitive foundation of human linguistic competence(Goldstein et al. 2022).

2 Related Work

So far, there are not many AI papers and datasets that focus specifically on Linguistic Olympiad problems. LINGOLY dataset is a challenging Linguistic Olympiad puzzle that includes over 90 low-resource languages(Bean et al. 2024). There also exists PuzzLing dataset, a Rosetta-Stone-style collection based on Linguistic Olympiad puzzles(Şahin et al. 2020).

Meanwhile, there is a prior work comparing the performance of the ToT technique versus Standard I/O using the PuzzLing dataset(Lin et al. 2023). A step presents three possible English translations for each input sentence and, at the next step, classify each translation's plausibility as Sure, Maybe, or Impossible. If a translation is judged Impossible, it returns to the previous step to regenerate interpretations. This study reported that applying ToT actually resulted in worse performance than Standard I/O, and hypothesized the following reasons:

1. Prompt Accuracy: The prompts were not precise enough, causing confusion in the LLM.
2. Sensitivity of the Evaluation Method: The current evaluation method may have failed

to properly detect contradictions introduced by new candidate solutions.

3. LLM Capability: Using GPT-3.5-Turbo may have lacked the performance capacity to handle the computational demands of ToT.
4. Fit of the ToT Architecture: ToT may not be well-suited for solving linguistic problems; in other words, global pattern analysis might be more important than partial problem solving when inferring language rules.

Therefore, this study seeks to analyze global patterns while reducing the number of iterations and input length compared to ToT.

3 Method

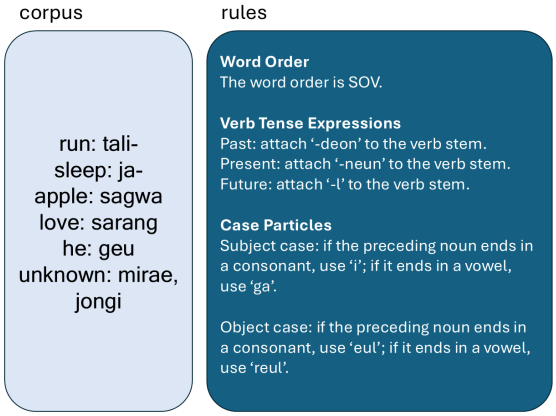

Figure 1: Example of corpus and rules. The corpus is formatted as "low-resource language: English."

This study introduces two elements, corpus and rules, for analyzing global patterns without iteration. The corpus organizes the meanings of words appearing in the example sentences, and the rules are a list of inference rules that adequately explain the given examples (Figure 1). Bean et al. (2024) calculated the validity of multiple possible translations, but this paper considers that approach somewhat arbitrary because it does not explicitly describe the derivation process. Therefore, this paper extracted a list of explicit rules. This paper also newly introduced the corpus element, which was absent in previous studies. This draws on human language usage patterns; unless near-native proficiency is required, vocabulary is crucial for language learning and sentence generation. Furthermore, because many pilot-test responses failed to fill in entire words, it was expected that providing a corpus matching word meanings would improve performance.

Next, to examine whether "iteration and updating" improve the rules and corpus, two architectures were designed. In this study, we will compare the performance of an iterative updating method—designed to reduce computation compared to ToT—with a non-iterative method. Through this comparison, we aim to determine whether the LLM is capable of updating its own outputs.

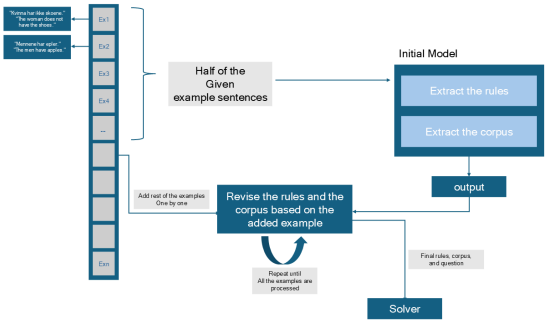

Figure 2: Iterative Architecture. The Initial Model takes half of the entire set of example sentences and extracts both rules and the corpus. It then combines this output with the next batch of example sentences and feeds them into the Grammar agent; this process repeats until all examples have been consumed. Once the iterations are complete, the Solver agent is provided with the target problem along with the accumulated rules and corpus, and is instructed to return the answer.

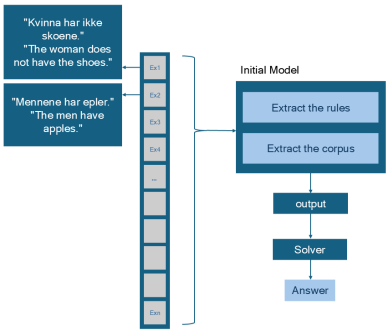

Figure 3: Non-iterative Architecture. The initial model takes the entire set of example sentences at once to extract both rules and outputs. It then delivers these results together with the problem to be solved to the Solver Agent and receives the answer.

4 Dataset

This paper uses PuzzLing dataset for analysis. The PuzzLing dataset is a small translation puzzle collection extracted from Linguistic Olympiad Rosetta Stone type problems, and it encompasses a variety of morphological, syntactic, and phonological features across 81 languages(Şahin et al. 2020). Each puzzle is designed to lead the solvers to infer the governing rules from only a minimal number of example sentences and translate new sentences accordingly, thereby requiring human-level reasoning ability and meta-linguistic awareness rather than statistical methods. Although the PuzzLing dataset is more challenging than the LINGOLY dataset, only 10 problems have publicly released answers, and performance on the remaining problems can be checked only via the dashboard. The results will be provided at the end.

5 Experiments

5.1 Experimental Settings

The code and prompts can be viewed at the GitHub link. Experiments were conducted via OpenAI API calls (temperature = 0.5). Unless otherwise specified, the model used was gpt-3.5-turbo; any use of gpt-4o is explicitly noted.

5.2 Evaluation Metrics

We used the four metrics provided on the PuzzLing dashboard (Exact match, BLEU-2, CTER, CHRF) unchanged.

6 Results

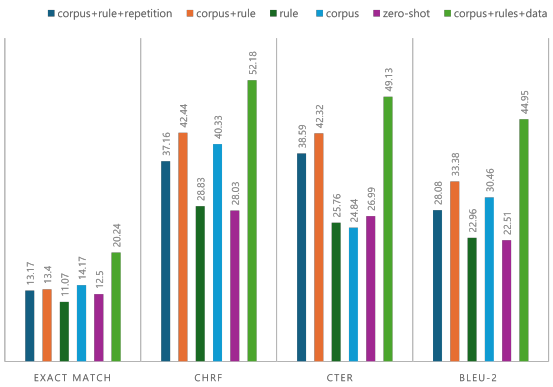

Figure 4: Exact match, CHRF, CTER, and BLEU-2 scores for five variations. Only the one labeled "repetition" corresponds to Architecture 1, i.e., the iterative version. All others have had repetition removed.

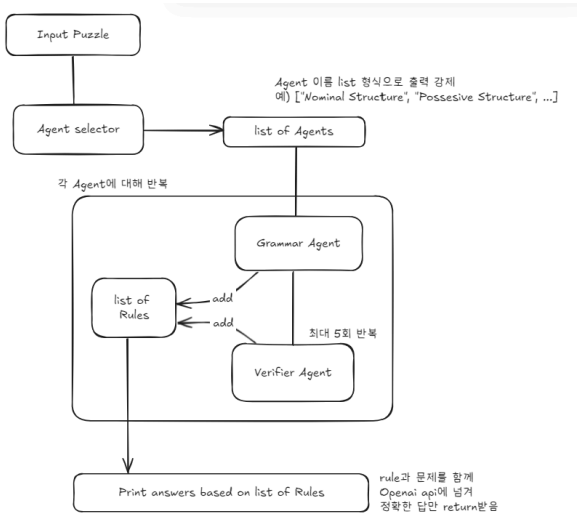

Figure 5: Diagram of the newly designed ToT structure. we added an Agent selector, which was not present in prior literature, and separated the Grammar Agent from a Verifier Agent that evaluates and refines its output(Lin et al. 2023). The role of the Agent selector in planning which Grammar Agent to deploy was crucial.

No.	exact match	bleu
1	0.00	5.19
2	85.71	94.15
3	12.50	17.68
4	0.00	1.97
5	10.00	19.21
6	0.00	18.53
7	0.00	10.99
8	0.00	14.13
9	42.86	67.59
10	37.50	63.32
average	18.86	31.28

Table 1: Experimental results of the ToT structure (Figure 5). It takes more than ten times longer, but the results did not improve proportionally to the time invested.

6.1 Ineffectiveness of the Tree-of-Thoughts Approach

Table 1 shows the outcome of the implementation in Figure 1, which was ultimately discarded and not discussed in detail in the main text. Unlike Figure 2 and Figure 3, this implementation included an "Agent selector" component that determines which grammatical elements are required to solve the problem. In terms of Exact match and BLEU-2 scores, it does not outperform the structure without repetition. We hypothesize the

following reasons for its failure despite consuming significantly more computational resources:

1. The Agent selector is not accurate. To select the correct Agent, one must first understand the grammatical requirements; however, to understand those requirements, one must already have selected the correct Agent, resulting in a circular paradox.
2. The Verifier agent is unable to actively refine the output of the Grammar agent. This is discussed in more detail in section 6.2.

6.2 Limitations in LLM Self-Revision and Verification

Modifying the examples or guidelines in the prompt did not lead to any improvement of prior outputs. Prior research has shown that without external input, it is difficult for an LLM to self-correct its own outputs(Huang et al. 2024). The failure of both the ToT structure and the Iterative Structure to improve performance in the Linguistic Olympiad Solver supports this claim.

Even though the Verifier agent and the Grammar agent are invoked separately with completely different prompts, the model was barely able to revise its own outputs. This was true not only for the ToT structure but also for standard iterative processes.

As shown in Figure 4, the results of iterative validation using both corpus and rules (Figure 2) are worse than those obtained by removing iteration but still using corpus and rules (Figure 3). Except for the case where only rules are introduced after removing repetition, the performance of the repetition variant is lower than that of the comparison groups.

6.3 Greater Impact of Corpus Extraction over Rule Extraction

In the Non-iterative architecture, the effect of corpus on performance improvement was greater than that of rules. As shown in Figure 4, the rule-only variant does not display a significant difference from zero-shot. Instead, the impact of the corpus—which was assessed as less important than rules—proved to be substantial. Corpus analysis shows particular strength in ChrF, an N-gram-based metric. One can attribute the success in placing words correctly to the assistance provided by the corpus.

6.4 In-Context Learning: Synergy between Corpus and Example Sentences

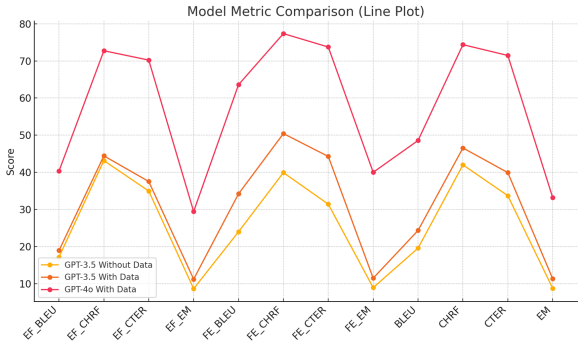

Figure 6: Actual scores as evaluated on the Dashboard. "EF" denotes English⟶Foreign-language, and "FE" denotes Foreign-language⟶English translation. Overall, FE performance exceeded EF, and the gain from adding example sentences was steeper for the FE task than for the EF task.

In the original Solver prompt, the example sentences provided in the problem were replaced by the previously obtained corpus and rules. The sections labeled "with-data" in the figure refer to the cases where the original example sentences were retained and provided alongside the corpus and rules. Although the corpus and rules explicitly list the inferred grammatical patterns and vocabulary extracted from the examples, the AI seems to gain further ideas when it is given not only the corpus and rules but also the original example sentences. Thus, it appears that the combination of example sentences with the corpus and rules boosts in-context learning. The fact that performance is higher when example sentences and the corpus & rules coexist implies a synergy between the two.

7 Conclusion

This study compared multiple prompting architectures—including an iterative updating method, a non-iterative method, and a Tree-of-Thoughts variant—on PuzzLing dataset. The results show that the ToT approach not only failed to outperform simpler architectures but also consumed a lot more computation without meaningful gains. Furthermore, removing iteration while retaining both corpus and rule information (the non-iterative architecture) exceeded the performance of the iterative version, suggesting that repeated self-revision by the LLM provides little added benefit.

Corpus extraction proved more beneficial than explicit rule extraction, especially under the ChrF metric, indicating that vocabularies aid accurate translation more than formalized inference rules. Finally, it was demonstrated that providing example sentences together with the corpus and rules yields synergy in in-context learning: models achieve their highest accuracy when both raw examples and distilled knowledge co-occur. These findings suggest that LLMs excel when given concrete lexical examples and distilled patterns, but struggle with internal self-revision and abstract reasoning architectures like ToT. Future work should explore hybrid designs that combine example-driven prompting with simple pattern extraction while lowering computational overhead.

8 References

References

Bean, Andrew M., Simi Hellsten, Harry Mayne, Jabez Magomere, Ethan A. Chi, Ryan Chi, Scott A. Hale, and Hannah Rose Kirk. 2024. "LINGOLY: A Benchmark of Olympiad-Level Linguistic Reasoning Puzzles in Low-Resource and Extinct Languages". 2024. https://arxiv.org/abs/2406.06196.

Goldstein, A., Z. Zada, E. Buchnik, M. Schain, A. Price, B. Aubrey, S. Nastase, et al. 2022. "Shared Computational Principles for Language Processing in Humans and Deep Language Models".*Nat Neurosci.*

Huang, Jie, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. "Large Language Models Cannot Self-Correct Reasoning Yet". 2024. https://arxiv.org/abs/2310.01798.

Lin, Zheng-Lin, Chiao-Han Yen, Jia-Cheng Xu, Deborah Watty, and Shu-Kai Hsieh. 2023. "Solving Linguistic Olympiad Problems with Tree-of-Thought Prompting". In *Proceedings of the 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)*, edited by Jheng-Long Wu and Ming-Hsiang Su, 262–69. Taipei City, Taiwan: The Association for Computational Linguistics, Chinese Language Processing (ACLCLP). https://aclanthology.org/2023.rocling-1.33/.

Şahin, Gözde Gül, Yova Kementchedjhieva, Phillip Rust, and Iryna Gurevych. 2020. "PuzzLing Machines: A Challenge on Learning From Small Data". In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, edited by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, 1241–54. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.115.