Could language models win the International Linguistics Olympiad?

Jamie Garnham
Monash University
jgar0047@student.monash.edu

Ehsan Shareghi
Monash University
ehsan.shareghi@monash.edu

Abstract

Linguistic puzzles, wherein the solver must deduce rules of an unfamiliar language purely in-context, represent a uniquely perplexing problem format even for state-of-the-art large language models. Yet by exploring various inference-time scaling methods, we demonstrate that language models' performance on these problems can be improved without the need for fine-tuning or providing supplementary linguistic context. To this end, this paper introduces the first domain-specific inference-time scaling framework for linguistic puzzles, which we use to improve the performance of three model families - R1 (Deepseek), Gemini 2.5 Flash (Google), and Llama 3.3 70B Instruct (Meta) - on a challenging Linguistics Olympiad-based benchmark by 4.9, 13.1, and 4.9 percentage points, respectively. Nonetheless, even when multiple optimisations are applied, we find that LLMs' linguistic puzzle performance remains well below comparable mathematical and commonsense benchmarks, and we speculate as to why linguistic reasoning continues to pose a distinctive challenge for even the most capable large language models.[1]

1 Introduction

In 2025, OpenAI and Google announced that their large language models (LLMs) would have received gold medals in that year's International Mathematics Olympiad (IMO), a global mathematics competition for secondary students, by answering five of the six competition questions correctly (Huang and Yang, 2025). This is reflective of the growing reasoning capability of state-of-the-art LLMs, which have been shown to perform extremely well on arithmetic and coding benchmarks (Guo et al., 2025; Comanici et al., 2025). This strong performance on these benchmarks has been aided further by the development of inference-time scaling methods, where the models are allocated greater time and computational resources to produce outputs, allowing the models to self-correct, check for self-consistency, and hence improve performance (Muennighoff et al., 2025; Wang et al., 2023).

Contrastingly, LLMs perform significantly worse on benchmarks based on the International Linguistics Olympiad (IOL), the IMO's linguistic counterpart, designed to test secondary students' ability to recognise and apply patterns in real-world, often lesser-known languages (Chi et al., 2024; Bean et al., 2024). In light of this, there has been growing interest in measuring and improving the ability of LLMs to solve these problems, across the various puzzle formats present in the Linguistics Olympiad papers (see Figure 1 and Appendix A for examples of these formats) (Zhu et al., 2025; Choudhary et al., 2025).

This naturally leads to the question of whether inference-time scaling methods, which have seen success in other domains but are as yet largely untested in the linguistic puzzle space, can provide similar value when applied to this benchmark (Snell et al., 2025; Wang et al., 2023; Yao et al., 2023). Accordingly, we aim to optimise the 'extraction' of reasoning from these models, to hence determine whether their poor performance is remediable at inference-time, or is reflective of a true deficiency in the specialised reasoning modes required for this domain.

To test this, we comprehensively analyse the effect of scaling inference-time budget, measuring the performance of seven different inference-time scaling methods (including combinations thereof) across four different puzzle formats in the linguistic puzzle domain. We use a downscaled version of the LINGOLY-TOO dataset, intended to mirror this benchmark in difficulty, and compare three model

[1] Code and dataset available at https://github.com/JamieGarnham/lingoly-team and https://huggingface.co/datasets/jamiegarnham/lingoly-team

families: DeepSeek R1, Gemini 2.5 Flash, and Llama 3.3 70B Instruct (Guo et al., 2025; Comanici et al., 2025; Dubey et al., 2024).

From these experiments, we find that sampling multiple responses from an LLM in parallel and applying self-consistency to these outputs provides a moderate improvement above the baseline for all models, suggesting that there is value in scaling the models' inference-time budget to more accurately measure their capability in this domain. Conversely, we find that LLM-as-a-Judge-based approaches are ineffective in this problem domain, in large part because the processes of solving and correctness verification are not strictly separated for this problem format, unlike in mathematical or commonsense problems (Gu et al., 2024).

Finally, we design and implement a novel inference-time scaling and prompting framework, LINGOLY-TEAM, which isolates the individual reasoning steps and then aggregates an answer from a 'team' of LLM instances working in parallel. Our domain-specific framework outperforms all other approaches for the Llama model (scoring 17.7%, compared to a baseline of 12.8%), and provides significant improvement for the Gemini model (to 36.5%, from a baseline of 23.4%), but offers less advantage to the already-deliberative R1 model (34.8% compared to a 29.9% baseline). Despite still performing reasonably poorly overall, we find that in the Linguistics Olympiad puzzle domain, the more powerful R1 and Gemini 2.5 Flash models show reasoning capability *occasionally*; but to be considered true linguistic reasoners, it is necessary to develop the capacity of LLMs to do this *consistently*.

2 Background

Modern LLMs have become increasingly 'multilingual', enabled by efforts to increase the number of languages included in their pre-training data (Huang et al., 2024); state-of-the-art LLMs have also demonstrated the capacity to 'learn' a language purely in-context, when presented with a bilingual dictionary, small bilingual corpora, and explicit grammar rules (Tanzer et al., 2024; Court and Elsner, 2024; Zhang et al., 2024; Merx et al., 2024). Contrastingly, Linguistics Olympiad puzzles, which involve deducing a low-resource language's grammar and semantic rules only from a very limited set of paired examples and then applying a combination of these to a set of unseen test

Preamble: "Beja" is the Arabic name for the language which calls itself...

Context:
a. ilaga diwiini The male calf is sleeping.
b. doobaab rhitni She sees a bridegroom.
...

Questions:
3.1. Translate into English:
1. uukaam ootak rhaabu.
...
3.2. Translate into Beja:
6. A man meets the mouse.
...

Figure 1: 'Rosetta Stone' linguistic puzzle (source: United Kingdom Linguistics Olympiad 2013, Round 2, Question 3: Beja).

examples, constitute a challenging problem format for LLMs (Bozhanov and Derzhanski, 2013; Bean et al., 2024; Chi et al., 2024).

Methods to improve LLMs' ability to solve these linguistic reasoning problems include manually synthesising 'step-by-step' examples to guide LLMs to the solution (Zhu et al., 2025), retrieving comparable examples from languages of the same family (Ramji and Ramji, 2025) to augment the examples, and human-labelling individual morphemes in preprocessing (Choudhary et al., 2025), all of which result in an improvement in solution quality. Yet the performance reported in previous papers may not reflect LLMs' true linguistic reasoning capability for two key reasons: knowledge contamination (as these benchmarks consist of publicly available problem sheets), and reliance on external assistance (the supplementation of additional linguistic context by a human annotator). The former has been addressed by the recent release of an obfuscated Linguistics Olympiad benchmark, LINGOLY-TOO (Khouja et al., 2026) — on which LLM performance is found to be notably worse — yet the latter remains largely unexamined.

As the paucity of linguistic puzzles generally precludes fine-tuning, the logical approach to optimising LLMs' linguistic reasoning performance is inference-time scaling, wherein increased computational effort is invested in the generation and evaluation of a model's output, rather than in its training (Muennighoff et al., 2025; Snell et al., 2025).

One such method is self-consistency (Wang et al., 2023), whereby a single LLM is prompted many times to solve the same problem, and the most 'consistent' answer from the sample pool is taken, generally using majority voting, weighted voting, or an LLM-as-a-Judge approach (Wang et al., 2023; Gu et al., 2024). Despite their suitability for this problem, there has been no comprehensive attempt to apply inference-time scaling methods to the linguistic puzzle domain.

Furthermore, LLMs demonstrate sensitivity to the structure of the prompt, and investment in this has been shown to dramatically improve performance in other reasoning domains; most notably, chain-of-thought prompting encourages step-by-step reasoning (Wei et al., 2022), while Step-Back prompting guides the model to first reason about the principles required to solve the problem before answering it (Zheng et al., 2024). Providing LLMs with a generic linguistic puzzle-solving algorithm has shown promise in very small-scale studies (Khouja et al., 2026), but has yet to be thoroughly examined. Given the absence of systematic optimisation efforts in this domain, the upper bounds of LLMs' end-to-end linguistic reasoning remain undetermined.

3 LINGOLY-TEAM: Methodology

3.1 Dataset

The dataset used in this paper is an abridged version of the LINGOLY-TOO benchmark, which consists of Linguistic Olympiad problems across four formats: 'Rosetta Stone', 'Pattern', 'Match-up', and 'Monolingual' (Khouja et al., 2026). Each of these puzzles is a real-world Linguistics Olympiad problem with a systematic orthographic obfuscation (i.e. a swapping of vowels and consonants) applied to it, such that the language tested in the puzzle is 'unrecognisable' to an LLM and hence mimics the conditions of a human participant in a Linguistics Olympiad competition. Due to cost constraints associated with using proprietary models, we take a representative sample of the original dataset, filtering this benchmark to include exactly one randomly selected obfuscation per problem (out of the six obfuscations for each problem in the benchmark). The resulting dataset consists of 82 problem sheets that span 80 languages, containing a total of 173 questions, and 1,005 individual question parts. See Appendix A for examples of each problem format.

3.2 Baseline

After curating this dataset, the initial sample of 32 responses to each question for each model is generated, using the same baseline prompt as the LINGOLY-TOO benchmark (see Appendix B for the prompt used) (Khouja et al., 2026). A variation of this prompt is also tested, but not found to improve performance (see B for this ablation study).

To be consistent with the LINGOLY-TOO benchmark, exact matching is used to evaluate the correctness of each response (see Appendix C for corrigenda of the puzzle solutions), with no partial marks for semi-correct answers. Each model's temperature parameter is set to 1.0 to produce a diverse sample of responses, enhancing the effectiveness of self-consistency (Wang et al., 2023). Furthermore, from manual inspection, it is found that 46 of the 82 problem sheets contain arbitrarily ordered tables of language data (such as sets of paired vocabulary words). In these cases, these tables are randomly shuffled row-wise between runs to mitigate LLMs' sensitivity to the order in which information is presented (Pezeshkpour and Hruschka, 2024), and further promote diversity in the sample pool.

The baseline is taken to be the average exact match % across the sample of 32 responses for each model to minimise the variance of the baseline, given the stochasticity of model outputs. For all experiments, the reported accuracy is the % of subquestions where the LLM's output exactly matches any correct answer.

3.3 Inference-time budget simulation

Random subsamples of 16, 8, 4, 2, and 1 responses respectively are taken sequentially from this initial sample of 32. This subsampling simulates the effect on LLM performance of varying inference-time budget; an LLM that can generate and aggregate from a sample of 32 responses has a relative inference-time budget that is 32 times greater than an LLM that may only generate 1 response per question.

To determine the *upper bound* on the performance of the methods tested in this experiment, we also determine the proportion of samples that contain at least one correct answer at each subsample size. This upper bound is not considered to be a valuable indicator of model performance, rather as a trajectory against which to evaluate the

effectiveness of the various aggregation methods.

3.4 Aggregation methods

3.4.1 Self-consistency (majority voting)

For the first aggregation method, we apply self-consistency to each subquestion, whereby the most frequently occurring response from each sample is taken as the final answer for each subquestion from that sample. In cases where there is a tie for the 'majority' answer, random tiebreaking is used, to ensure that we have exactly one 'final answer' per problem. We use majority voting rather than weighted voting due to the unreliability of self-reported LLM 'confidence' scores (Yang et al., 2025), and the unavailability of token probabilities for the proprietary models.

3.4.2 LLM-as-a-Judge

Two distinct LLM-as-a-Judge approaches are also applied to all non-unanimous cases. The first is a 'reranking judge', whereby the LLM is provided with the problem sheet context, the question, and the set of all unique answers produced by the same LLM; the model is then asked to rerank the answers according to quality from first to last, with the answer ranked first by the LLM selected as the judge's answer. The rationale behind this method is to reduce open-answer questions (which comprise the majority of subquestions in the dataset; see Experiments 4) to multiple choice questions, or to narrow the set of options available in the case of multiple choice questions, with the aim of increasing the likelihood that the aggregated answer is the correct one. The second approach ('top-1 judge') aims to simplify the judging task, prompting the LLM to select only the single 'best' output, rather than (largely arbitrarily) reranking all outputs. See Appendix D for the used LLM-as-a-Judge prompts.

3.4.3 Hybrid vote/judge

To strike a balance between minimising the likelihood of an errant judge contradicting a large majority, and ensuring that slim 'pluralities' are not overvalued in cases where there is significant division among LLMs, we propose a hybrid approach; if the size of the majority (the frequency of the most common answer in a sample) exceeds a 50% threshold, the majority answer is used; otherwise, the judge's answer is used as the final response. This method is tested separately for both the reranking judge and the top-1 judge.

We compare the accuracy of each of these five aggregation methods for each model and subsample size.

3.5 LingOly-Team

3.5.1 Induction-application with self-consistency

A unique prompting method, combining elements of Chain-of-Thought and Step-Back prompting, is also developed for each problem format. From our own analysis, we find that the 'Rosetta Stone', 'Pattern', and 'Monolingual' puzzles can generally be solved using the following two-step approach:

1. **Induction:** Determine the language's rules (lexical semantics, syntax, and morphology) from the provided examples
2. **Application:** Use analogy to apply these rules to the test example

Accordingly, we propose a new framework, adapted for the linguistic puzzle domain:

1. Prompt the model to first determine as many 'rules' of the language as possible and write these out (but not answer any questions). Record this output. See Appendix E for the used prompt.

2. In the same context window, prompt the model to use the rules it has determined to answer the first question. See Appendix E for the used prompt.

3. Repeat **2.** for each subsequent question in the problem sheet.

This approach is intended to force the model to 'step back' and examine the whole problem sheet, rather than take shortcuts in answering the questions based on only a partial reading of the provided examples, or repeatedly change its interpretation of the language's rules after having already answered previous subquestions (both of which are observed behaviours in the preceding experiments). We design variants of this for each puzzle format, tailored to the typical characteristics of each problem type (see E).

For the 'Rosetta Stone', 'Pattern', and 'Monolingual' questions, we repeat this induction-application prompting approach to generate a sample of 32 sets of 'thoughts' and consequent responses for each problem sheet. From this sample and each constituent subsample (of sizes 16, 8, 4, 2, and 1), we then aggregate the final answer from the sample using majority voting.

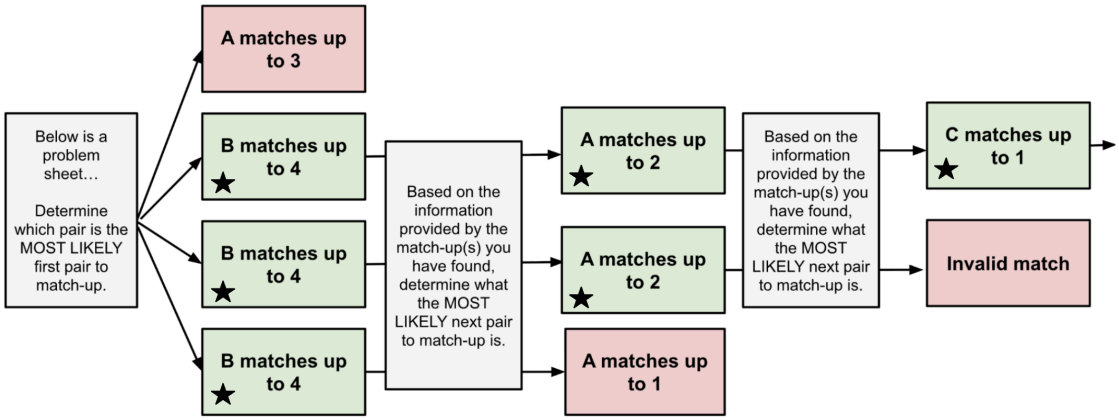

Figure 2: Tree-of-Matches (for a subsample of size 4). Boxes marked with ⋆ represent the selected response at each stage.

3.5.2 Match-up format: Tree-of-Matches

The 'Match-up' puzzle format is unique in that it is not solved using the same induction-application algorithm as the other formats; because the provided examples are unpaired, there is initially no paired data from which to induce the rules of the language. The general algorithm for these problems is instead as follows:

1. Determine the 'most likely' initial pairing of examples (this may be based on a loanword, or some other context clue).
2. Based on the initial pairing determined, deduce some set of rules for the language.
3. Based on the existing set of rules, determine the next 'most likely' pairing of examples. If there are multiple possible options, keep track of all 'paths'.
4. Based on the new pairing determined, add any new rules to the existing set of rules.
5. Repeat **3.** and **4.** until each example has been paired.

We design a framework to prompt the model to follow this algorithm sequentially, and we take the final answer to be the most frequent match-up at each step; this is summarised in Figure 2. See Appendix E for the used prompts.

4 Experiments and analysis

4.1 Baseline performance

Figure 3 shows the baseline performance for each model. The DeepSeek R1 model is the most effective baseline model, scoring 29.9 (σ=1.25), followed by the Gemini 2.5 Flash model, with a baseline score of 23.4 (σ=1.40); the Llama 3.3 70B Instruct model is the lowest scorer at 12.8 (σ=0.95).

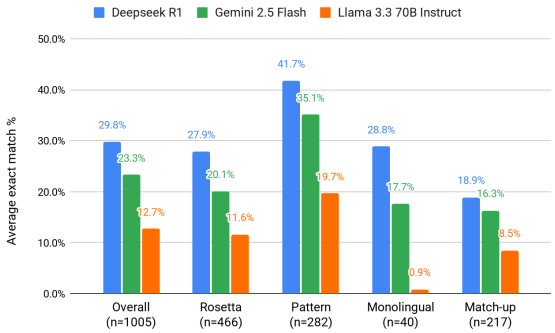

Figure 3: Baseline performance by puzzle format (average exact match % across a sample of 32 responses to each subquestion).

As in previous benchmarks, all three models perform best on the simpler 'Pattern' problems, while the complex 'Match-up' puzzles are the most challenging for the R1 and Gemini 2.5 Flash models. The relatively lightweight Llama 3.3 70B model performs exceptionally poorly on the 'Monolingual' puzzles, potentially because these puzzles all involve numerical reasoning as well as linguistic reasoning.

The DeepSeek R1 and Llama 3.3 70B models both perform better here than in the original LINGOLY-TOO benchmark (29.9 vs. 26.5 and 12.8 vs. 8.2 respectively). This is attributed to the introduction of shuffling of the tabular question data (see 3.2), which mitigates LLMs' sensitivity to row order, and the use of only one obfuscated version of each problem (as opposed to the six versions used in the original benchmark), which may introduce some stochasticity (see 3.1). Given these solving conditions differ slightly, all comparisons in this paper use our own baseline to ensure the

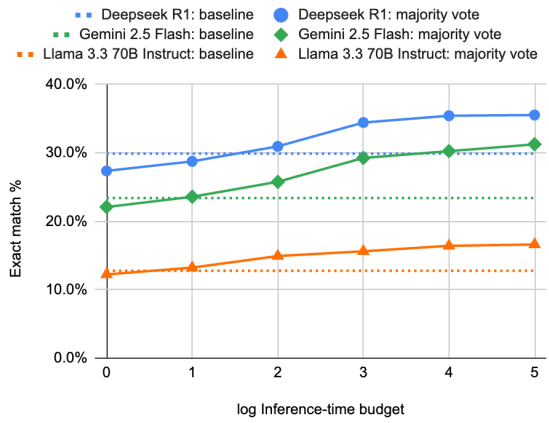

Figure 4: Inference-time budget vs. majority vote accuracy.

validity of results.

4.2 Aggregation methods

4.2.1 Self-consistency

As Figure 4 illustrates, increasing the inference-time budget indeed improves performance for all models when self-consistency is applied, with benefits plateauing above sample sizes of 16. With majority voting across 32 samples, benchmark scores increase to 35.5 (R1), 31.2 (Gemini 2.5 Flash), and 16.6 (Llama 3.3), representing gains of 5.6, 7.8, and 3.8 percentage points, respectively, compared to the baseline.

This method proves most effective for the speed-optimised Gemini 2.5 Flash model, as majority voting mitigates the impact of the model's tendency to take 'shortcuts' (evidenced by its high variance of accuracy). Notably, this enables the faster Gemini model to outperform R1's baseline, indicating that for linguistic puzzles, allocating inference-time budget to repeated sampling may be more effective than using a single sample from a slower-reasoning model.

4.2.2 LLM-as-a-Judge

Figure 5 summarises the effectiveness of each aggregation method tested. The upper bound trajectory represents a theoretical 'perfect' judge that is credited with a correct answer whenever *at least one* correct answer is present in the sample. This upper bound significantly exceeds both baseline and majority vote performance, providing a key insight: the models can produce correct responses for many subquestions, but only occasionally, insufficiently frequently to form a majority.[2]

However, LLM-as-a-Judge methods are found to be largely ineffective. Reranking, top-1, and hybrid judge approaches all perform worse than majority voting across all models, with the reranking method failing to exceed even the random baseline for Gemini. The reranking task proved too complex, with models frequently producing unparsable outputs or failing to follow instructions. While the top-1 judge and hybrid methods showed minor improvements over reranking, neither consistently beat majority voting.

4.3 Induction-application prompting with self-consistency

As shown in Figure 6, the induction-application method near-ubiquitously improves the baseline performance for all three models across the three applicable puzzle formats. While improvements are minor for R1 on Rosetta Stone and Pattern puzzles (likely because this model already uses a similar method without needing explicit prompting), this approach yields an 18% point gain on Monolingual puzzles, suggesting that format-specific algorithms can effectively boost linguistic reasoning performance, even for slow-reasoning models.

Combined with self-consistency, the effects compound substantially. Performance improves with larger samples, plateauing around n=8 for Pattern and Monolingual formats. For Rosetta Stone puzzles, gains reach 7.8, 13.8, and 2.8 percentage points for R1, Gemini 2.5 Flash, and Llama 3.3 70B respectively. Gemini benefits most, outperforming even R1 on two formats, indicating that algorithmic guidance and increased inference-time budget can unlock this speed-optimised model's reasoning capability beyond what its baseline performance suggests.

4.4 Tree-of-Matches

The results of applying our Tree-of-Matches method to the fourth and most challenging problem format, the Match-up puzzles, are shown in Figure 7. Performance generally improves with increasing inference-time budget, though much less monotonically; this instability is attributed to the complexity of our sequential prompting framework, which results in a high rate of invalid or unparsable responses (12.0%, 12.4%, and 27.2%

[2]This strong upper bound performance is only partially attributed to the 26.5% of subquestions that are essentially 'multiple choice'.

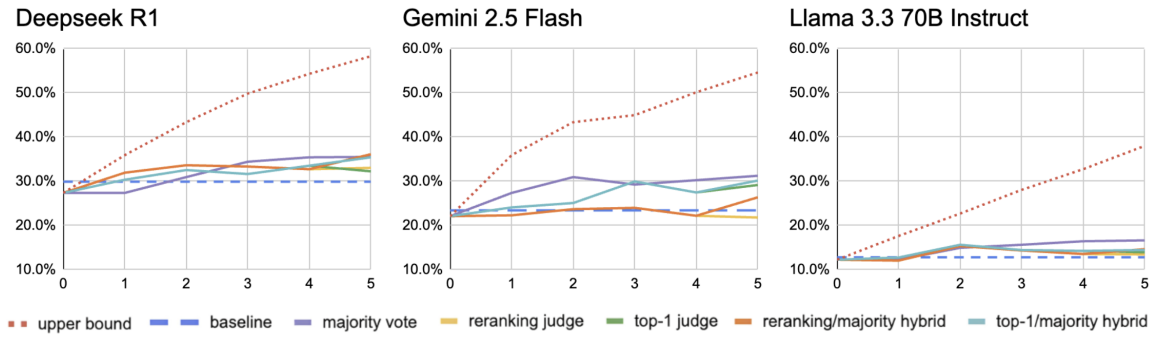

Figure 5: Aggregation methods vs. theoretical upper bound.

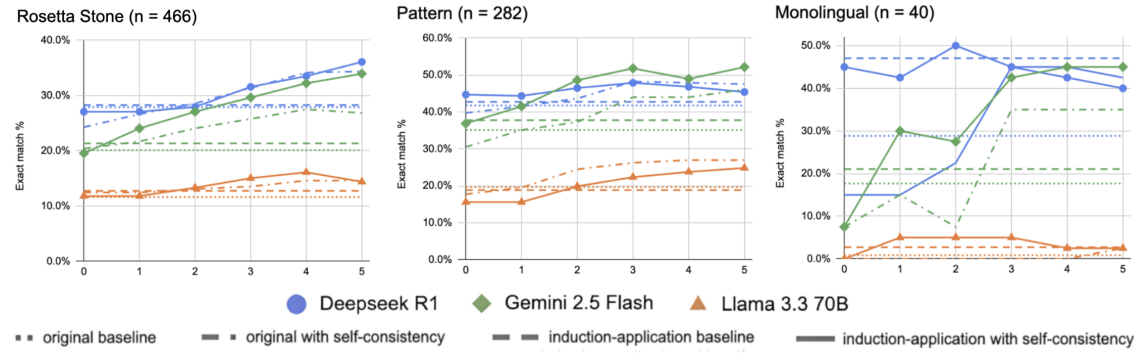

Figure 6: Induction-application with self-consistency.

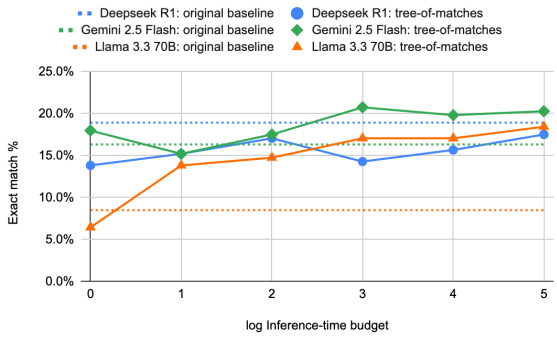

Figure 7: Tree-of-Matches performance.

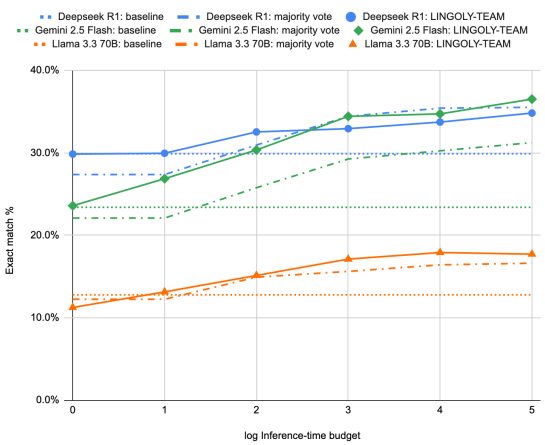

Figure 8: LINGOLY-TEAM vs. baseline with original prompting framework, and majority voting with original prompting framework.

for R1, Gemini, and Llama respectively). Conversely, the Llama model's performance improves by 9.9 percentage points over the baseline to 18.4%, nearly matching R1's performance. Inspection of outputs suggest that while our algorithm helps the models determine the first few matches (the 'low-hanging fruit'), the later, more difficult matches remain out-of-reach, even for the most powerful models. This puzzle format remains a largely unsolved challenge; further iteration on this method is left for future work.

4.5 LINGOLY-TEAM: Overall Performance

By combining the results of our two format-specific algorithms with inference-time scaling, we obtain a set of results for all puzzles in the filtered benchmark, summarised in Figure 8. Overall, optimal inference time-scaling offers significant improvement over the baseline for all three models. For the R1 model, our format-specific algorithm performs

marginally worse than self-consistency with the original prompting framework, attributed largely to this model's poorer performance on the Match-up puzzles using this method. Conversely, for Gemini 2.5 Flash and Llama 3.3 70B, the LINGOLY-TEAM method outperforms all other approaches. The Gemini model in particular improves markedly using this method to a score of 36.5%, an increase of 5.3 percentage points above any other method, and 13.1 points above its baseline performance, exceeding even the much more powerful R1 model's best score (36.0%), suggesting that this method may be best suited to boosting the performance of faster, mid-sized models.

5 Discussion

The performance of our LINGOLY-TEAM method, particularly the increase of the speed-optimised Gemini 2.5 Flash model's performance above that of the slower-reasoning R1 model, suggests that dividing inference-time budget across parallel instances and aggregating an output can be more effective than simply scaling the allocated 'reasoning time' for a single instance, mirroring the improvement seen in other domains (Snell et al., 2025). Additionally, for some linguistic puzzle formats, extrapolation of the models' accuracy curve to larger sample sizes shows potential for further performance improvement, indicating that there may be benefit to scaling inference-time budget even beyond the sample sizes tested here.

The failure of all LLM-as-a-Judge methods tested likely stems from the nature of linguistic puzzles: the tasks of solving the puzzle and verifying a solution's correctness are inextricable. Reasoning traces suggest the judge must solve the problem independently before selecting an answer, hence a 'judge' offers little advantage over a single sample. The effectiveness of a 'judge' LLM in this domain could be improved by instead prompting the 'judge' to evaluate intermediate reasoning steps, rather than simply the final answer; this is left for future work.

Notably, for each of the three models, the number of subquestions answered correctly at least occasionally is almost twice the number of subquestions answered correctly a majority of the time. This suggests that most of the questions are not 'impossible' for LLMs; their issue is with finding consistency, a central aspect of becoming 'true' logical reasoners. In the case of the R1 'reasoning' model, this inconsistency suggests some deficiency of the model's purported self-correction capacity, which, if improved, could reduce the frequency of flawed reasoning pathways and hence increase performance on similar benchmarks.

Ultimately, our experiments demonstrate that LLMs have significant room for improvement in the modes of reasoning required for these linguistic puzzles. These include subword and character-level reasoning in puzzles where morphemes are not clearly delineated, which has been shown to pose challenges for LLMs that tokenise at a word or subword-level (Shin and Kaneko, 2024). Furthermore, some of the puzzles require phonological reasoning - the ability to recognise similar-sounding morphemes that may not necessarily be spelled identically - which has been shown to be challenging for text-based models (Suvarna et al., 2024). Another distinguishing feature of Linguistics Olympiad problems is context length; compared with their Mathematics Olympiad counterparts, linguistic puzzles involve a significantly longer initial context, and may include many semi-connected subquestions connected to a single problem context. Although state-of-the-art LLMs are able to retrieve information from increasingly long contexts, this performance degrades rapidly with increasing task complexity (Li et al., 2024); we speculate that when the problem is sufficiently intricate, even a medium-length context suffices to overwhelm the LLM and produce logical failures.

6 Conclusion

Adapting existing inference-time scaling methods to the linguistic puzzle problem domain proves effective in improving the benchmark performance of a variety of models, suggesting that when allocated a sufficient inference-time budget, the 'linguistic reasoning' ability of LLMs may exceed that suggested by existing linguistic puzzle benchmarks.

Even with significant investment in inference-time optimisation and a markedly increased compute budget, the performance of all the models on this benchmark remains well below a human-performance upper bound. We hence suggest that there is need for the development of LLMs' capacity in reasoning modes that are often underrepresented by mathematical or commonsense benchmarks, including longer-context, phonological, and character-level reasoning, in order for these models to be truly labelled as 'reasoners'.

Limitations

One obfuscation per problem: Unlike the original LINGOLY-TOO benchmark, which includes up to six obfuscations per problem, our dataset is limited to only one randomly selected obfuscation per problem, as taking repeated samples from proprietary models becomes prohibitively expensive over the entire benchmark. Although different obfuscations of the same problem should theoretically be of equal difficulty, and hence our sample should be a fair representation of the benchmark, this may introduce some variance to the results. As such, all comparisons in this paper are with our own baseline, rather than the LINGOLY-TOO baseline; it is left for future work to test LINGOLY-TEAM on the full dataset.

Simulated inference-time budget: Simulating inference-time budget using number of samples is effective when comparing results *for a given model*, but does not reflect the true inference-time budget when comparing *across models*, as some models spend more time and/or tokens on reasoning for each sample. For example, of the models tested, the R1 model spends significantly more time reasoning for each problem, whereas the other two models generally output significantly more tokens, rendering the true 'inference-time budget' available to each model difficult to control for. Accordingly, the paper's focus is on improving the performance of each individual model, as this allows fair comparison of inference-time budget across sample sizes. To enable a fairer comparison *across models*, compute-normalisation could be applied, though this is left for future work.

Format-specific algorithms: For our LINGOLY-TEAM method, we provide the LLM with a format-specific algorithm to solve each puzzle; however, to be truly 'unassisted' and replicate the Olympiad solving conditions, the LLM must be able to recognise to which of the four formats the puzzle belongs, and hence select the correct algorithm. As these formats are generally distinct from each other and consistent across competition papers, this task is deemed sufficiently trivial to be excluded.

No 'script' puzzles: The dataset does not include problems that involve reading or writing non-Latin scripts, as this relies on multimodal capabilities, which are not common to all the LLMs tested. However, these problems are common in Linguistics Olympiad Papers, and so to truly test how well an LLM would perform in the International Linguistics Olympiad, these questions would need to be included; this is left for future work.

Exact matching only: All model responses in our experiments are evaluated using exact matching only, whereas in real-world Linguistics Olympiad competitions, partial marks are given for answers that are partially correct or show evidence of correct reasoning. With the recent release of human-annotated step-by-step solutions to linguistic puzzles (Lian et al., 2025), partial marking is now feasible; the use of these solutions to evaluate the induction phase of the LINGOLY-TEAM framework represents an unexplored but promising pathway for future work.

References

Andrew Michael Bean, Simeon Hellsten, Harry Mayne, Jabez Magomere, Ethan A Chi, Ryan Andrew Chi, Scott A. Hale, and Hannah Rose Kirk. 2024. LINGOLY: A benchmark of olympiad-level linguistic reasoning puzzles in low resource and extinct languages. In The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track.

Bozhidar Bozhanov and Ivan Derzhanski. 2013. Rosetta stone linguistic problems. In Proceedings of the Fourth Workshop on Teaching NLP and CL, pages 1–8, Sofia, Bulgaria. Association for Computational Linguistics.

Nathan Chi, Teodor Malchev, Riley Kong, Ryan Chi, Lucas Huang, Ethan Chi, R. McCoy, and Dragomir Radev. 2024. ModeLing: A novel dataset for testing linguistic reasoning in language models. In Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP, pages 113–119, St. Julian's, Malta. Association for Computational Linguistics.

Mukund Choudhary, KV Aditya Srivatsa, Gaurja Aeron, Antara Raaghavi Bhattacharya, Dang Khoa Dang Dinh, Ikhlasul Akmal Hanif, Daria Kotova, Ekaterina Kochmar, and Monojit Choudhury. 2025. UNVEILING: What makes linguistics olympiad puzzles tricky for LLMs? In Second Conference on Language Modeling.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and Chris Hahn. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. Preprint, arXiv:2507.06261.

Sara Court and Micha Elsner. 2024. Shortcomings of LLMs for low-resource translation: Retrieval and understanding are both the problem. In Proceedings of the Ninth Conference on Machine Translation, pages 1332–1354, Miami, Florida, USA. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 82 others. 2024. The llama 3 herd of models. CoRR, abs/2407.21783.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. 2024. A survey on llm-as-a-judge. CoRR, abs/2411.15594.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. Nature, 645(8081):633–638.

Kaiyu Huang, Fengran Mo, Hongliang Li, You Li, Yuanchi Zhang, Weijian Yi, Yulong Mao, Jinchen Liu, Yuzhuang Xu, Jinan Xu, Jian-Yun Nie, and Yang Liu. 2024. A survey on large language models with multilingualism: Recent advances and new frontiers. CoRR, abs/2405.10936.

Yichen Huang and Lin F. Yang. 2025. Winning gold at IMO 2025 with a model-agnostic verification-and-refinement pipeline. In The 5th Workshop on Mathematical Reasoning and AI at NeurIPS 2025.

Jude Khouja, Lingyi Yang, Karolina Korgul, Simeon Hellsten, Vlad A. Neacșu, Harry Mayne, Ryan Othniel Kearns, Andrew M. Bean, and Adam Mahdi. 2026. LINGOLY-TOO: Disentangling reasoning from knowledge with templatised orthographic obfuscation. In The Fourteenth International Conference on Learning Representations.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context llms struggle with long in-context learning. CoRR, abs/2404.02060.

Da-Chen Lian, Ri-Sheng Huang, Pin-Er Chen, Chunki Lim, You-Kuan Lin, Guan-Yu Tseng, Zhen-Yu Lin, Pin-Cheng Chen, and Shu-Kai Hsieh. 2025. LOBSTER: Linguistics olympiad benchmark for structured evaluation on reasoning. In Proceedings of the 37th Conference on Computational Linguistics and Speech Processing (ROCLING 2025), pages 193–229, National Taiwan University, Taipei City, Taiwan. Association for Computational Linguistics.

Raphaël Merx, Aso Mahmudi, Katrina Langford, Leo Alberto de Araujo, and Ekaterina Vylomova. 2024. Low-resource machine translation through retrieval-augmented LLM prompting: A study on the Mambai language. In Proceedings of the 2nd Workshop on Resources and Technologies for Indigenous, Endangered and Lesser-resourced Languages in Eurasia (EURALI) @ LREC-COLING 2024, pages 1–11, Torino, Italia. ELRA and ICCL.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candes, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. In Workshop on Reasoning and Planning for Large Language Models.

Pouya Pezeshkpour and Estevam Hruschka. 2024. Large language models sensitivity to the order of options in multiple-choice questions. In Findings of the Association for Computational Linguistics: NAACL 2024, pages 2006–2017, Mexico City, Mexico. Association for Computational Linguistics.

Raghav Ramji and Keshav Ramji. 2025. Inductive linguistic reasoning with large language models. In Findings of the Association for Computational Linguistics: ACL 2025, pages 22783–22810, Vienna, Austria. Association for Computational Linguistics.

Andrew Shin and Kunitake Kaneko. 2024. Large language models lack understanding of character composition of words. In ICML 2024 Workshop on LLMs and Cognition.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In The Thirteenth International Conference on Learning Representations.

Ashima Suvarna, Harshita Khandelwal, and Nanyun Peng. 2024. PhonologyBench: Evaluating phonological skills of large language models. In Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024), pages 1–14, Bangkok, Thailand. Association for Computational Linguistics.

Garrett Tanzer, Mirac Suzgun, Eline Visser, Dan Jurafsky, and Luke Melas-Kyriazi. 2024. A benchmark for learning to translate a new language from one grammar book. In The Twelfth International Conference on Learning Representations.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net.

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.

Daniel Yang, Yao-Hung Hubert Tsai, and Makoto Yamada. 2025. On verbalized confidence scores for LLMs. In ICLR Workshop: Quantify Uncertainty and Hallucination in Foundation Models: The Next Frontier in Reliable AI.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In Thirty-seventh Conference on Neural Information Processing Systems.

Chen Zhang, Xiao Liu, Jiuheng Lin, and Yansong Feng. 2024. Teaching large language models an unseen language on the fly. In Findings of the Association for Computational Linguistics: ACL 2024, pages 8783–8800, Bangkok, Thailand. Association for Computational Linguistics.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. 2024. Take a step back: Evoking reasoning via abstraction in large language models. In The Twelfth International Conference on Learning Representations.

Hongpu Zhu, Yuqi Liang, Wenjing Xu, and Hongzhi Xu. 2025. Evaluating large language models for in-context learning of linguistic patterns in unseen low resource languages. In Proceedings of the First Workshop on Language Models for Low-Resource Languages, pages 414–426, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Example Puzzles

Examples for Rosetta Stone, Pattern, Monolingual, and Match-up puzzles are provided in Figure 9.

B Ablation study - Reduced Context Prompt

We carry out a small ablation study to determine if reducing the complexity of the baseline prompt improves performance. The baseline prompt contains the entire set of questions in the problem sheet, and then asks the LLM to answer a specific question from this set. However, we hypothesise that this may 'overwhelm' the model, and cause confusion about which question it must answer. To test this, we remove any 'irrelevant' questions from the prompt, as illustrated in Figure 10.

We test this with the DeepSeek R1 model only, and perform self-consistency on subsamples of size 16, 8, 4, 2, and 1. This performance is compared to the 'original' baseline prompt in Figure 11.

We see that while the 'reduced context' prompt results in a marginally improved baseline, as inference-time budget increases, it is outperformed by the original prompt. This suggests that for some problem sheets, the other questions provide some information that helps the model answer the given question. Additionally, allowing the model to 'see' all questions before answering a given question better replicates the solving conditions of human participants in Linguistics Olympiad competitions, as all questions in a problem sheet are visible to the participant at all times. Accordingly, this 'reduced context' prompt is abandoned for the remainder of this project, in favour of the original baseline prompt from the LINGOLY-TOO benchmark.

C Manual corrections to dataset

For all questions, correct answers are evaluated case-insensitively. Occasionally the set of correct answers will be missing trailing full stops, despite these being present in the exemplars, leading the LLMs to include them; to fix this, full stops are always stripped from both the LLM output and the correct response when evaluating correctness. See Table 1.

D LLM-as-a-Judge prompts

Puzzle-specific prompts used for LLM-as-a-Judge methods are provided in Figure 12 and Figure 13.

E LINGOLY-TEAM Prompts

Puzzle-specific prompts used for LINGOLY-TEAM are provided in Figure 14-17.

F Full Results

Full results for DeepSeek R1, Gemini 2.5 Flash, and Llama 3.3 70B Instruct are provided in Table 2.

Rosetta Stone Puzzle (Beja — UKLO 2013, R2Q3)

Preamble: "Beja" is the Arabic name for the language which calls itself "ti bedawye", the unwritten language...

Context:

a. ilaga diwiini — The male calf is sleeping.
b. doobaab rhitni — She sees a bridegroom.
c. gwibu — It is a mouse
...

Questions:

3.1. Translate into English:
1. uukaam ootak rhaabu.
2. faar katamya.

...

3.2. Translate into Beja:
6. A man meets the mouse.
7. The bridegroom is not eating

...

Pattern Puzzle (Dinka — UKLO 2022, R2Q4)

Context:

1st person	3rd person	Translation
nàaň	nòǒň	to have
kwòoc	kùuc	to not know
...		

Question:

4.2) Assuming that the following verbs conform to the previous pattern, fill in the correct form on your answer sheet:

1st person	3rd person	Translation
lwòǒǒj	b)	to be different
d)	cèěm	to eat
...		

Monolingual Puzzle (Gumatj — UKLO 2019, R1Q8)

Context:

1) lurrkun rulu ga wanggang + wanggang rulu ga wanggang = dambumiriw rulu ga marrma
2) lurrkun + lurrkun rulu ga lurrkun = dambumiriw rulu ga wanggang
...

Question:

Q.2. Write the following Gumatj numbers in Arabic numerals.
6) wanggang
7) dambumiriw rulu ga lurrkun

...

Match-up Puzzle (Albanian — UKLO 2023, R1Q6)

Context:

Below are some questions in Albanian, in a random order, and their English translations, in alphabetical order. Note that ë is a vowel and ç is a consonant.

1. Pse është në Angli? — a) Did you drink anything?
2. Kujt ia shiti? — b) Did you kill someone?
3. Kë vrau? — c) How did you dance?

...

Question:

Q 6.1 Match the Albanian sentences to their English translations.
Follow-up question(s):
Q 6.2 Translate:
(a) Ku kërceu?

...

Figure 9: Examples of linguistic puzzles used in Olympiad-style problems: (1) Rosetta Stone translation (Beja), (2) morphological Pattern discovery (Dinka), (3) Monolingual numerical reasoning (Gumatj), and (4) bilingual Match-Up (Albanian).

Instructions: Below is a problem sheet . . .

Problem sheet context: Here are a number of sentences in Language X. . .

~~**All questions**: Q1. Translate from English to Language X. . . Q2. Translate from Language X to English. . .~~

Specific question: Now answer the following question:
Q1. Translate from English to Language X. . .

Figure 10: Baseline prompt sketch, from the LINGOLY-TOO benchmark. The strikethrough text is omitted from the 'reduced' prompt.

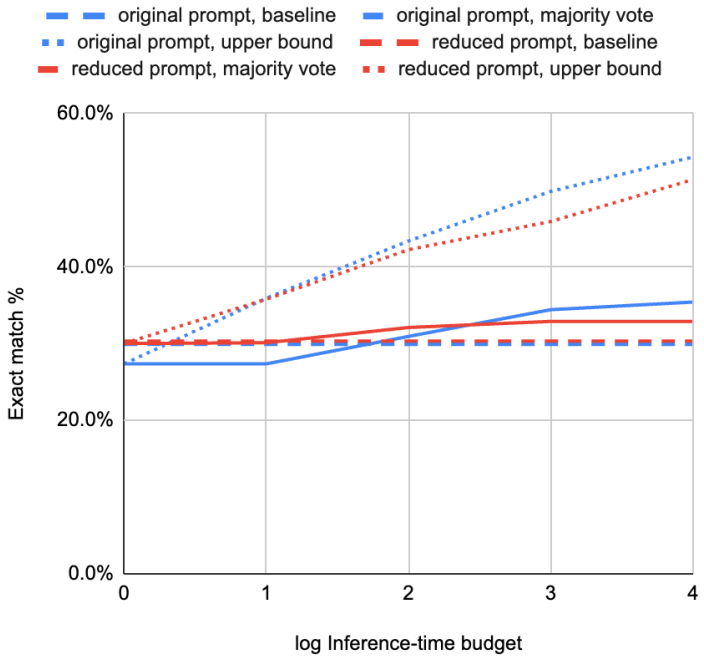

Figure 11: Comparison of original and reduced prompt performance for all puzzle formats (using the DeepSeek R1 model).

Correction 1: Problem sheet 5, Q5.1(a) *Before:* "sopostüd", "üpgontüd" *After:* Any string containing both "sopostüd" and "üpgontüd" *Explanation:* Questions asks for two words but does not specify order/format.
Correction 2: Problem sheet 75, Q7(3) *Before:* "(2n)" *After:* "(2n)" or "two people who are not siblings" *Explanation:* Question explains "(2n)" is an abbreviation of "two people who are not siblings", but does not specify that the abbreviation must be used.
Correction 3: Problem sheet 170, Q5(k) *Before:* "langgbu'" *After:* "langgbu'" (note different apostrophe) *Explanation:* LLMs output ' (U+2019), but answer expects ' (U+0027), considered equivalent.

Table 1: Corrections made to solutions in the dataset.

Judge prompt: ‘Evaluate the following solutions to a linguistic puzzle and return ONLY a JSON dictionary with scores.’
Puzzle context: ‘PUZZLE: [Main question text and specific subquestion extracted from the problem sheet]’
Possible answers: ‘POSSIBLE ANSWERS:’
‘1. [First unique answer from model responses]’
‘2. [Second unique answer from model responses]’
‘3. . . . ’
Scoring instructions: ‘Evaluate each answer’s correctness and assign a ranking to each one, where 1 is the answer that you think is most likely to be correct, 2 is the next most likely, etc. Consider: whether each answer could satisfy the puzzle constraints, the possible logical reasoning behind each answer, and how well it addresses what the puzzle is asking.’
Output format: ‘Output ONLY a valid JSON dictionary in this exact format:’ ‘{"answer_1": 0.0, "answer_2": 0.0, . . . }’

Judge response: ‘{"answer_1": 1.0, "answer_2": 2.0, "answer_3": 3.0, . . . }’

Figure 12: Reranking judge prompt sketch.

Preamble: ‘Below is a problem sheet from a linguistics exam. You will first see the entire sheet, then be asked to respond to a specific subquestion from the sheet. You will be given a set of options to choose from. Your answers to the questions should rely only on reasoning about the information provided in the sheet.’
Problem sheet: ‘[Full cleaned problem sheet, including language data and examples]’
Subquestion: ‘Now provide the answer to the following subquestion. [Extracted question header, e.g. “Translate into Language X:”] [Subquestion identifier and content, e.g. “a. the blind milkman”]’
Options: ‘These are the options you have to choose from:’
‘[Option 1]’
‘[Option 2]’
‘[Option 3]’
‘. . . ’
Instructions: ‘Consider the logic that could be used to lead to each of the options presented. Based on your reasoning, please select one option only that you think is the correct answer. If you think multiple options could be correct, select only one of them.’
Output format: ‘Your output MUST end with a valid JSON dictionary in this exact format:’ ‘{"answer": "option"}’

MC Judge response: ‘[Reasoning. . .] {"answer": "[Selected option]"}’

Figure 13: Top-1 judge prompt sketch.

Induction prompt: ‘Below is a problem sheet from a linguistics exam. Your task is to determine as much information about the language as possible, purely from the information provided. You should systematically go through the information provided, and try to determine the vocabulary meaning of each word (where possible), the syntactic structure (such as word order), the morphology (including any verb conjugations), and the meaning of any affixes or subwords. Test every piece of information you determine against every example provided.’

Problem sheet context: ‘Here are a number of sentences in Language X. . . ’

Model response: ‘Let’s systematically analyse this language...’

Application prompt 1: ‘Based on the information about the language you have determined, solve the following puzzle:’

Question: ‘3.1. Translate into English:’

Model response: ‘{"a": "The cow sits..."}’

Application prompt 2: ‘Based on the information about the language you have determined, solve the following puzzle:’

Question: ‘3.2. Translate into Language X:’

Model response: ‘{"a": "uutook gwib..."}’

Figure 14: Induction-application prompt sketch for a ‘Rosetta Stone’ puzzle. (adapted from: United Kingdom Linguistics Olympiad 2013, Round 2, Question 3: Beja).

Induction prompt: ‘Below is a problem sheet from a linguistics exam. Your task is to determine as much information about the language as possible, purely from the information provided. You should systematically go through the information provided, and try to determine the morphological and phonological patterns of the language (such as noun declension), including the meaning of any subwords or affixes you may see. Look for systematic patterns in how the language forms syllables, words, and phrases. Test every piece of information you determine against every example provided.’

Problem sheet context:

1st person	3rd person	Translation
nàaň	nòǒň	to have
kwòoc	kùuc	to not know
...		

Model response: ‘MORPHOLOGICAL ANALYSIS OF LANGUAGE X:’

Application prompt: ‘Based on the patterns you have identified in the language, solve the following puzzle:’

Question: ‘4.2) Assuming that the following verbs conform to the previous pattern, fill in the correct form on your answer sheet...’

Model response: ‘a) kwuc...’

Figure 15: Induction-application prompt sketch for a ‘Pattern’ puzzle (adapted from: United Kingdom Linguistics Olympiad 2022, Round 2, Question 4: Dinka).

Induction prompt: ‘Below is a problem sheet from a linguistics exam. Your task is to determine as much information about the language and its number system as possible, purely from the information provided. You should systematically go through the information provided, and try to determine the vocabulary meaning of each number, the base of the number system (e.g. decimal, hexadecimal), the syntactic structure (such as word order), the morphology, and any other patterns you can see in the language’s number system. Test every piece of information you determine against every example provided.’

Problem sheet context:
1) lurrkun rulu ga wanggang + wanggang rulu ga wanggang = dambumiriw rulu ga marrma
2) lurrkun + lurrkun rulu ga lurrkun = dambumiriw rulu ga wanggang
...

Model response: ‘Let’s analyse the number system of Language X’

Application prompt: ‘Based on the information about the language you have determined, solve the following puzzle:’

Question: ‘Q.2. Write the following Language X numbers in Arabic numerals.’

Model response: ‘6) 3...’

Figure 16: Induction-application prompt sketch for a ‘Monolingual’ linguistic puzzle (source: United Kingdom Linguistics Olympiad 2019, Round 1, Question 8: Gumatj).

Instructions: Below is a problem sheet from a linguistics exam. Your answers to the questions should rely only on reasoning about the information provided in the sheet.

Context: Below are some questions in Language X, in a random order...

1. Pse është në Angli?
2. Kujt ia shiti?
3. Kë vrau?

a) Did you drink anything?
b) Did you kill someone?
c) How did you dance?

Initial match-up question:
Your task is to:
1. Determine which pair is the MOST LIKELY first pair to match-up.
2. Express this match-up using the following JSON: {{"%%": "X"}}
where %% is the serial, and X is the corresponding translation.
Do not match-up any other pairs yet.

Initial model response: {"1": "c)"}

Next match-up question:
Now let's suppose that the following information is correct:
1 matches up to c)
1. Based on the information provided by the match-up(s) you have found, determine what the MOST LIKELY next pair to match-up is...

Follow-up question instructions: Based on the linguistic patterns and correspondences you have identified, answer the following question:

Follow-up question: Q 6.2 Translate: (a) Ku kërceu?...

Figure 17: Induction-application prompt sketch for a 'Match-up' linguistic puzzle (source: United Kingdom Linguistics Olympiad 2023, Round 1, Question 6: Albanian).

Sample size	Majority vote	Upper bound	Rerank judge	Top-1 judge	Rerank/ majority hybrid	Top-1/ majority hybrid	LINGOLY-TEAM
DeepSeek R1 (baseline: 29.9)							
1	27.4	27.4	27.4	27.4	27.4	27.4	29.9
2	27.4	35.9	31.9	30.3	31.9	30.3	30.0
4	30.9	43.4	33.6	32.5	33.6	32.5	32.5
8	34.4	49.9	33.3	31.6	33.3	31.6	32.9
16	35.4	54.3	32.7	33.5	32.7	33.5	33.7
32	35.5	58.3	33.0	32.2	36.1	35.4	34.8
Gemini 2.5 Flash (baseline: 23.4)							
1	22.1	22.1	22.1	22.1	22.1	22.1	23.6
2	27.4	35.9	22.3	24.1	22.3	24.1	26.9
4	30.9	43.4	23.7	25.1	23.7	25.1	30.3
8	29.3	45.0	24.0	30.0	24.0	30.0	34.4
16	30.2	50.1	22.2	27.5	22.2	27.5	34.7
32	31.2	54.6	21.8	29.2	26.4	30.1	36.5
Llama 3.3 70B Instruct (baseline: 12.8)							
1	12.2	12.2	12.2	12.2	12.2	12.2	11.2
2	12.2	17.6	12.0	12.7	12.0	12.7	13.1
4	14.9	22.7	15.3	15.6	15.3	15.6	15.1
8	15.6	28.1	14.3	14.4	14.3	14.4	17.1
16	16.4	32.7	13.5	14.2	13.5	14.2	17.9
32	16.6	38.0	13.4	13.9	14.6	14.4	17.7

Table 2: Comparison of methods across models. Section headers separate model families and report their respective baselines.