Could language models win the International Linguistics Olympiad?

James Garnham
Monash University
jgar0047@student.monash.edu

Ehsan Shareghi
Monash University
ehsan.shareghi@monash.edu

Abstract

Linguistic puzzles, such as those used in International Linguistics Olympiad competition papers, are a uniquely challenging problem format for large language models, testing the solver's ability to deduce and apply complex rules of an unfamiliar language from only a small number of examples. When forced to rely only on in-context reasoning, rather than 'memorised' knowledge, even state-of-the-art LLMs have been shown to perform poorly on these problems; yet we suggest that existing linguistic puzzle benchmarks may understate the true reasoning capability of these LLMs. By exploring various inference-time scaling methods, we demonstrate that models' performance on these problems can be improved without the need for fine-tuning or providing the models with additional information. To this end, we introduce LINGOLY-TEAM, a novel approach that involves providing a 'team' of up to 32 parallel LLM instances with a domain-informed algorithm to solve the puzzles, then aggregating the set of responses they produce to obtain a final answer. With this method, we improve the performance of three model families - DeepSeek R1, Gemini 2.5 Flash, and Llama 3.3 70B Instruct - on a challenging linguistic puzzle benchmark by 4.9, 13.1, and 4.9 percentage points, producing final scores of 34.8%, 36.5%, and 17.7% respectively. Even when multiple optimisations are applied, LLMs' linguistic puzzle performance remains well below comparable mathematical and commonsense benchmarks, indicating that linguistic reasoning continues to pose a distinctive and unsolved challenge for even the most capable large language models.[1]

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

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 (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 §2.1 and §A for full descriptions and examples of these formats) (Zhu et al., 2025; Choudhary et al., 2025; Majmudar and Filatova, 2025).

Given the public availability of these competition problems, the validity of previous benchmarks in this domain is compromised by knowledge contamination, with models able to simply 'memorise' the answers from their pre-training. This issue has been mitigated by the recent release of the LINGOLY-TOO benchmark, which consists of unrecognisable but equally challenging obfuscations of these puzzles; this forces the LLM to apply linguistic reasoning to solve the problems, and hence better replicate the 'unseen' solving conditions of the IOL's student participants, who are not intended to have any proficiency in the languages tested (Khouja et al., 2025).

With a more reliable linguistic reasoning benchmark in place, 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 by up to 32 times, 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 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-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 introduce 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. This 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 evidence of using logic to reach correct answers occasionally; but to be considered true linguistic reasoners, it is necessary to develop the capacity of LLMs to do this consistently.

2 Related work

2.1 Linguistic reasoning problems

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). This capability is improved even further by the advent of LLMs with very large context size limits, giving them the potential to process and retrieve information from entire grammar textbooks and dictionaries within a single context window (Li et al., 2024b). What has proven consistently more challenging for LLMs is to deduce a language's grammar and semantic rules from a very limited set of paired examples, with little to no vocabulary items or explicit grammar rules provided, and then apply a combination of these to a set of unseen test examples. This constitutes the problem format central to the International Linguistics Olympiad and its regional equivalent competitions, known as a 'Rosetta Stone' or 'Rosetta' puzzle (Figure 1 illustrates an example[2]), and several benchmarks consisting of these puzzles have been developed (Bozhanov and Derzhanski, 2013; Bean et al., 2024; Chi et al., 2024).

Recently, there has also been increased focus on the ability of LLMs to solve other puzzle formats that make up the Linguistics Olympiad competition papers (Bean et al., 2024; Khouja et al., 2025). These include 'Pattern' puzzles, which generally involve more explicit identification and application of

[2]All example puzzles used in this paper are available via https://www.uklo.org/past-exam-papers/

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
...

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

a linguistic pattern, 'Monolingual' puzzles, which often involve determining words for numbers and some arithmetic in another language, and the most challenging format of 'Match-up' puzzles, where the solver is presented with a set of jumbled examples in 'Problemese' (the tested language) and 'Solverese' (the solver's native language) and must determine which corresponds to which; examples of these puzzle formats are shown in §A.

2.2 Linguistics Olympiad benchmarks

Various methods have been developed to improve LLMs' ability to solve these linguistic reasoning problems. These 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, 2024) 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. However, these approaches all involve human annotation, or rely on the retrieval of information about the specific language tested, resulting in conditions that may not replicate those of a 'human' team competing in the International Linguistics Olympiad without access to external resources (IOL Committee, 2024). Additionally, as these puzzles are publicly available online, knowledge contamination appears to be an increasingly prevalent concern; a significant portion of LLMs' performance on these puzzles has retrospectively been attributed to recognition of the puzzles and consequent 'memorisation' of the solutions by models, rather than reasoning in an unseen context as intended (Khouja et al., 2025). An obfuscated version of the benchmark, LINGOLY-TOO, has since been developed to address this, wherein the name and any identifying features of the language are redacted, while the puzzles themselves are orthographically obfuscated; this involves permuting the vowel and consonant sounds in the 'Problemese' examples and solutions (for example, 'omrelgi' may be permuted to 'amroldu') to produce unrecognisable but structurally identical problems. On this benchmark, which better reflects the 'unseen' solving conditions of Linguistics Olympiad competitions, LLMs perform poorly, below even an 'average' human baseline (Khouja et al., 2025) and likely well below state-of-the-art models' gold-medal level performance in the equivalent International Mathematics Olympiad (Huang and Yang, 2025)[3].

2.3 Inference-time scaling methods

In general, where models perform poorly on a problem type, fine-tuning is used to improve domain-specific capability. However, the linguistic puzzle domain presents a unique challenge in that these puzzles exist only in the hundreds, precluding the effective fine-tuning of reasoning models (Vieira et al., 2024); furthermore, synthesising further examples at scale is challenging due to the precisely contrived structure of the puzzles, generally requiring the input of human linguists (Majmudar and Filatova, 2025). In light of this, a suitable approach for improving LLM performance is inference-time (or test-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).

A simple but effective example of an inference-

[3]Gold medallists in IOL generally score 65-90% in the finals round, while the best-performing LLM, Gemini 2.5 Pro, scores just 27% on LINGOLY-TOO problems from Round 2 of the United Kingdom Linguistics Olympiad, which is one round before the international competition. However, the real-world IOL competition involves partial marking, whereas this benchmark does not, rendering it difficult to ascertain exactly how well an 'LLM participant' in the competition would perform.

time scaling method is self-consistency (Wang et al., 2023), whereby a single LLM is prompted many times to solve the same problem in such a way that it may take a diverse set of reasoning paths. This can be achieved by using the model's temperature parameter (which controls how deterministic the model's outputs are) or shuffling the order of information in the question (Pezeshkpour and Hruschka, 2024). Consequently, the most 'consistent' answer from the sample pool is taken, using majority voting, weighted voting, or another selection method (Wang et al., 2023). Another method of aggregating from a sample pool of responses is the LLM-as-judge approach (Gu et al., 2024) which involves asking a 'judge' or 'verifier' LLM to rerank the set of responses.

LLMs also demonstrate sensitivity to the structure of the prompt, and investment in this can dramatically improve performance. Most notably, chain-of-thought prompting encourages step-by-step reasoning (Wei et al., 2023), while Step-Back prompting (source: step-back source) guides the model to first reason about the principles required to solve the problem before answering it, and both these methods have been shown to produce performance improvements on mathematical and logical reasoning benchmarks. The Tree-of-Thoughts method combines this with inference-time scaling, and involves prompting LLMs to first give their intermediate 'thoughts' about a given problem, evaluating these 'thoughts', and then following only 'promising' reasoning paths to completion (Yao et al., 2023). As yet, there has been no attempt to apply inference-time scaling methods to linguistic puzzles, besides one case, where a Tree-of-Thoughts approach is applied to one of the four linguistic puzzle formats, the 'Rosetta Stone' puzzles, yet falls short of the baseline performance (Lin et al., 2023). However, given their proven effectiveness at improving performance on comparable reasoning benchmarks, inference-time scaling methods appear promising in this domain, warranting a comprehensive analysis of a greater variety of these methods across a greater diversity of problem formats.

3 LingOly-Team: Methodology

3.1 Dataset

The dataset used in this paper is a filtered version of the LingOly-TOO benchmark, which consists of orthographically obfuscated versions of Linguistic Olympiad problems across four formats: 'Rosetta', 'Pattern', 'Match-up', and 'Monolingual' (see §2.1 and §A for format descriptions and examples) (Khouja et al., 2025). Although the original benchmark consists of multiple obfuscations per problem, this is filtered to exactly one randomly selected obfuscation per problem, as having the same exact solution for each LLM attempt is necessary to test self-consistency and aggregate answers across multiple attempts. The resulting dataset consists of 82 problem sheets that span 80 languages, containing a total of 173 questions, and 1,005 individual question parts. For 3 question parts out of 1,005 total, we manually adjust the set of correct answers to resolve ambiguity in the answer format expected by the prompt, hence preventing substantially correct answers errantly being labelled as 'incorrect' (these corrections are explained in §B).

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 (a variant of this prompt was also tested; see §C) (Khouja et al., 2025). A sketch of the baseline prompt is shown in Figure 2.

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…

Figure 2: Baseline prompt sketch, from the LingOly-TOO benchmark

Exact matching is used to evaluate the correctness of each response. Although partial marks are allocated in real-world Linguistic Olympiad competitions, exact matching is deemed the most appropriate evaluation metric for these experiments, as automatic partial correctness metrics such as chrF do not capture the domain-specific knowledge used to inform the (human-judged) partial marking of Linguistic Olympiad competition papers (Bean et al., 2024), and using a human evaluator is infeasible for these experiments given their scale.

Each model's temperature parameter is set to 1.0 to increase the likelihood of generating a diverse sample of responses to most questions: a necessary prerequisite for the subsequent aggregation methods to be effective (Wang et al., 2023). Additionally, we manually inspect each problem sheet, and find that 46/82 problem sheets contain tabular data whose order does not significantly affect the difficulty of the questions; in these cases, we randomly shuffle the rows of the tabular data 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 calculated by taking the average exact match % across the sample of 32 responses for each model; this baseline is used, as opposed to taking the exact match % of a single run, to minimise the variance of the baseline, given the stochasticity of model outputs.

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 to come.

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 consistent' or 'majority vote' (i.e. 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. By taking this 'majority vote' from each subsample size, we can analyse the effect of increasing the simulated inference-time budget on the average accuracy of the 'most frequent' answer.

We also compare the distribution of the frequency of the majority answer, where the majority is correct, where the majority is incorrect, and where no answer is correct. This is done to provide insight into the models' behaviour; whether their responses are sufficiently diverse, whether they consistently or occasionally arrive at the correct answer, and whether they are 'confidently wrong' (i.e. large majority size) or more sporadic when they are incorrect.

3.4.2 Output reranking judge

Two distinct LLM-as-judge approaches are also applied to each subquestion, in all cases where there is some 'dispute' about the answer (i.e., there is more than one distinct answer in the subsample); samples where all the answers are correct, or all are incorrect, are excluded from this method, as it is arbitrary to judge between identically correct responses. For the first of these judge approaches, the same LLM used to generate the responses is provided with the problem sheet context, the question, and the set of all unique answers to the question that are present in the sample; this model is then asked to rerank the answers according to quality from first to last. The answer ranked first by the LLM is finally selected as the 'judged' answer. Although the 'judge' is the same model used to generate the samples, the rationale behind this method is to reduce open-answer questions (which comprise the majority of subquestions in the dataset; see §4.2.2) to multiple choice questions, or to narrow the set of options available for the remaining multiple choice questions.

3.4.3 Top-1 judge

Considering only the top-ranked answer is used, the second, simpler judge approach involves prompting the model only to select the best answer out of the options taken from the sample ('top-1 judge'). This is intended to reduce the complexity of the problem for the model to solve, and hence ensure all the 'reasoning' time is devoted to finding the best response, rather than on arbitrarily ranking lower-quality responses. Sketches of the prompts used for the two judging methods are shown in Figure 3.

3.4.4 Hybrid vote/judge

To strike a balance between minimising the likelihood of an errant judge contradicting a large majority, and ensuring that slim 'majorities' 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.

Finally, 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', 'Pattern', and 'Monolingual' puzzles are generally 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.

2. In the same context window, prompt the model to use the rules it has determined to answer the first question.

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

Figure 4 illustrates this framework for 'Rosetta' problems.

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). Minor variations of this prompt are used for each puzzle format, tailored to the typical characteristics of each problem type (see §D).

For the 'Rosetta', '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.

3.5.2 Measuring 'promising' thoughts

By separating the inductive reasoning stage from the analogistic application stage, we analyse whether 'promising' reasoning paths tend to result in correct answers, and hence determine if the models' poor performance on this benchmark is likely due to poor inductive reasoning, or poor application of this reasoning (Qiu et al., 2024). This insight into the models' reasoning process could be used as the basis of a process verification mechanism; however, due to the significant manual effort required to annotate and obfuscate explanations of solutions to all the problems, this is left for future work.

Instead, we employ a novel approach. We perform a logistic regression with mixed effects on the full sample for each of the three puzzle formats tested, to determine whether a model answering the first subquestion in a problem sheet correctly - considered a proxy for the model taking a 'good' reasoning path - is a predictor of the model answering subsequent subquestions correctly, while controlling for the variation in difficulty across problem sheets. From this, we seek to determine whether correct answers cluster together - indicative of a logical reasoner - or appear independent of each other - indicative of a potentially 'illogical' reasoner.

3.5.3 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

Reranking judge

Instructions: "Evaluate the following solutions to a linguistic puzzle..."

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: Q1a"

Options: "POSSIBLE ANSWERS:
1. beddi and rırdı
2. beddi and gomıstı
..."

Task: "Evaluate each answer's correctness and assign a ranking to each one..."

Top-1 judge

Instructions: "...You will be given a set of options to choose from..."

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: Q1a"

Options: "These are the options...
beddi and amroldu
beddi and gomıstı
..."

Task: "...please select one option only that you think is the correct answer..."

Figure 3: Comparison of the prompts for the reranking judge, and the top-1 judge

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.

As the 'Match-up' problems involve a much larger number of reasoning steps (equal to the number of pairings to be made, up to 30 in this dataset), and early errors generally sabotage the chance of future pairings being correct, we develop a method for this problem format based on the Tree-of-Thoughts approach, which we call 'Tree-of-Matches'. With this method, the model is provided with the problem sheet, and prompted to determine the 'most likely' initial pairing; then, based on the additional information that this pairing provides, the most likely next pairing; then the next, and so on, until all items have been matched-up. This is repeated to produce a sample of 32 parallel 'branches' (which is then subsampled, as described previously). For each subsample size, at each 'step' the most consistent response is recorded, and the 'paths' corresponding to it are then followed; at the next step, this repeats, until the most 'consistent' answer at each step has been determined. The Tree-of-Matches approach is summarised in Figure 5 (a full prompt sketch is provided in §D).

4 Experiments and analysis

4.1 Baseline performance

Figure 6 shows the baseline performance for each model calculated by taking the average exact match % across the 32 runs for each model at a subquestion level (overall n = 1005 subquestions). 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), while the Llama 3.3 70B Instruct model is the lowest scorer at 12.8 (σ=0.95). As is seen in the LingOly-TOO benchmark, 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

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 4: Induction-application prompt sketch for a 'Rosetta' puzzle

Flash models. The relatively lightweight Llama 3.3 70B model performs exceptionally poorly on the 'Monolingual' puzzles, which all involve numerical reasoning as well as linguistic reasoning.

The two models that are also included in the LINGOLY-TOO benchmark (DeepSeek R1 and Llama 3 70B Instruct) both perform better here than in the original benchmark (29.9 vs. 26.5 and 12.8 vs. 8.2). Besides sample size, the most probable cause of this appears to be the introduction of shuffling of the tabular question data, which may mitigate LLMs' sensitivity to the order in which information is presented (and hence prevent the models falling into the same incorrect behaviour pattern across obfuscations of the same problem), or the use of only one obfuscation per problem (as opposed to six in the original dataset), which may introduce some stochasticity. Given these solving conditions differ slightly, all comparisons in this paper use our own baseline to ensure the validity of results.

4.2 Aggregation methods

4.2.1 Self-consistency

As Figure 7 illustrates, for all tested models, increasing the simulated inference-time budget available to each model results in improved performance when self-consistency is applied to the model outputs, but this benefit generally appears to plateau for sample sizes above 16, suggesting that a sample size of 16-32 may optimise the tradeoff between performance and compute budget. By taking the 'majority vote' of a sample of 32, the three models' benchmark scores increase to 35.5 (R1), 31.2 (Gemini 2.5 Flash), and 16.6 (Llama 3.3), representing increases from the baseline by 5.6, 7.8, and 3.8 percentage points respectively. This suggests that random shuffling of the tabular question data, as well as introducing stochasticity to the model outputs using the temperature parameter, leads the model to take diverse reasoning pathways, resulting in a sufficiently diverse sample pool for self-consistency to be effective.

Self-consistency is most effective for this benchmark when applied to the powerful but speed-optimised Gemini 2.5 Flash model, suggesting that while this model has the 'reasoning capability' to solve many of the puzzle, it often takes 'shortcuts' to quickly produce an output (also showing the highest variance of accuracy across the 32 samples), a behaviour whose impact which can be

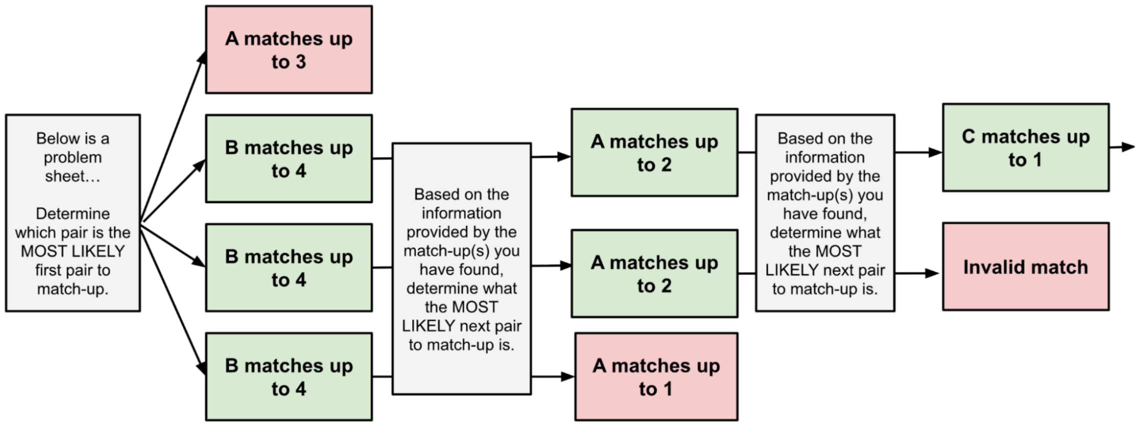

Figure 5: Tree-of-Matches (for a subsample of size 4)

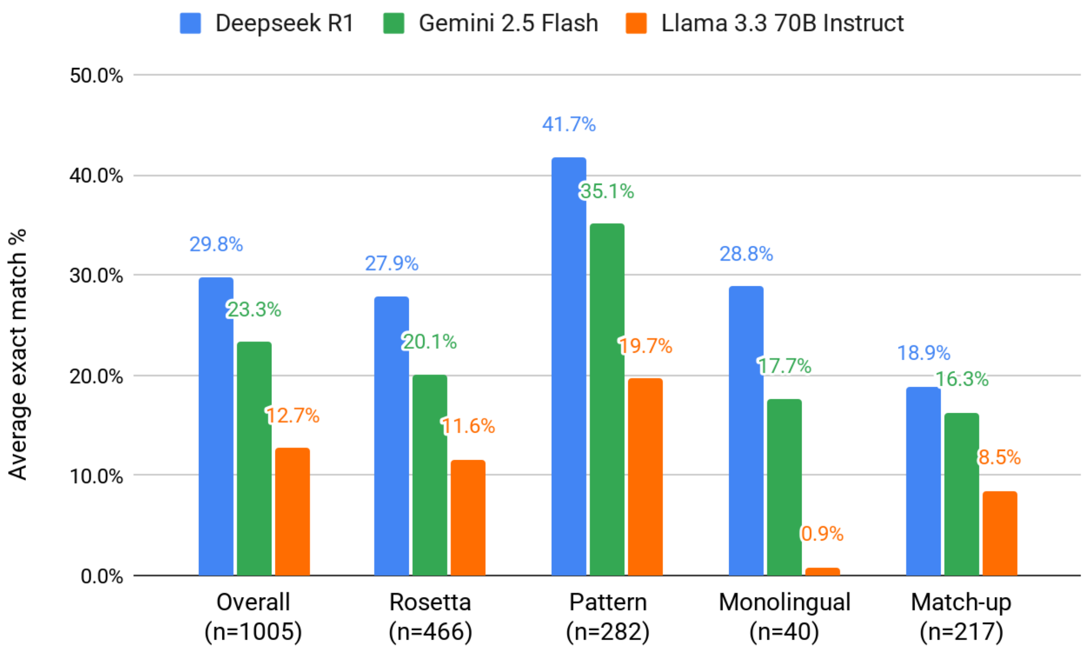

Figure 6: Baseline performance across puzzle formats (average from sample of 32)

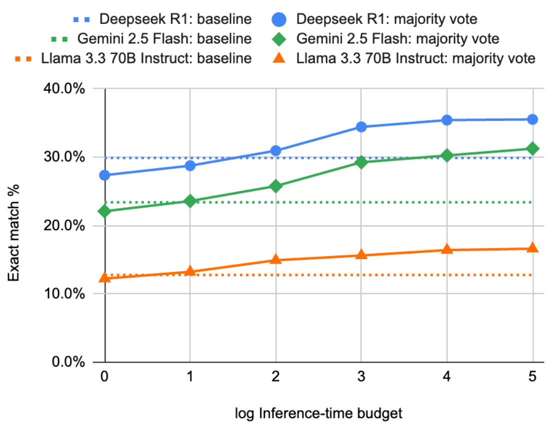

Figure 7: The effect of increased inference-time budget (simulated by sample size) on majority vote accuracy; each baseline is the average accuracy across the full sample of 32 (all logarithms are base 2 unless otherwise specified)

mitigated through majority voting. Notably, self-consistently enables the much faster Gemini model to outperform the baseline of the 'slow-reasoning' R1 baseline, suggesting that in the linguistic puzzle domain, allocating the inference-time budget to repeated sampling and self-consistency may be more effective than simply increasing the reasoning time.

To assess the diversity of the answers produced by each model within the sample of size 32, the distribution of the 'majority size' for each subquestion (the frequency of the most common answer) was also analysed (Figure 8). The R1 model is the most likely to be 'confidently correct', frequently producing the correct response all 32 times, while the Gemini model is the most likely to be 'confidently wrong' (i.e. produce the incorrect response every time). The Llama model appears to demonstrate the most stochasticity in its outputs, with the model rarely producing the same response to a given subquestion more than >25% of the time. For challenging subquestions - i.e. those that a model never answers correctly - the models tend to produce diverse outputs, evidenced by the often small majority size for these subquestions, suggesting that these models are taking a variety of reasoning pathways when challenged. For 'doable' questions - i.e. those that a model answers correctly at least some of the time - a 'two-thirds' majority tends to indicate that this majority answer is correct for the R1 model, whereas for the Llama and Gemini models, a significant majority may not be indicative of a correct answer.

4.2.2 LLM-as-Judge

Summary Figure 9 summarises the effectiveness of each of the five aggregation methods tested in this paper. The upper bound trajectory represents the performance of a theoretical 'perfect' judge - one that is able to select a correct answer from a sample whenever one is present, while the baseline accuracy is equivalent to using a 'random judge' that selects uniformly from the full sample (as the baseline is the average accuracy from the full sample).

Upper bound ('perfect judge') The trajectory of the upper bound curve shows that as the sample size increases, the probability that there is at least one correct answer in the sample increases approximately logarithmically, and for all three models, the upper bound significantly exceeds the performance of the baseline and the 'majority vote'. This provides a key insight in itself: that the models are able to produce the correct response for many of the subquestions, but only produce these correct answers occasionally (insufficiently frequently to form a majority).

To determine the extent to which multiple choice questions influence this (as even a random model will answer a multiple choice question correctly at least once if given sufficient attempts), we perform a manual analysis of all problem sheets included in the dataset. From this, we deem 26.5% (266/1005) of subquestions to have a finite set of answer 'options', either implicitly or explicitly, indicating that the high upper bound accuracy is only partially attributable to the presence of multiple choice questions[4].

Reranking judge Overall, the reranking judge method is not effective, performing worse than the majority vote approach for all three models, and failing to even exceed the 'random judge' baseline for the Gemini model. Particularly for lightweight models, the task of reranking up to 32 responses to an already-intricate puzzle is an extremely complex one, leading to an often seemingly random assignment of ranks.

The complexity of this task is evidenced by the models' observed tendency not to follow the task instructions, which are to assign a rank (i.e. 1 for the best, then 2, 3, etc.) to each response; instead, they frequently produce an unparsable output or

[4]For example, a question that presents a ten-syllable word, then asks for the 'number of the syllable on which emphasis should be placed', has ten implicit options.

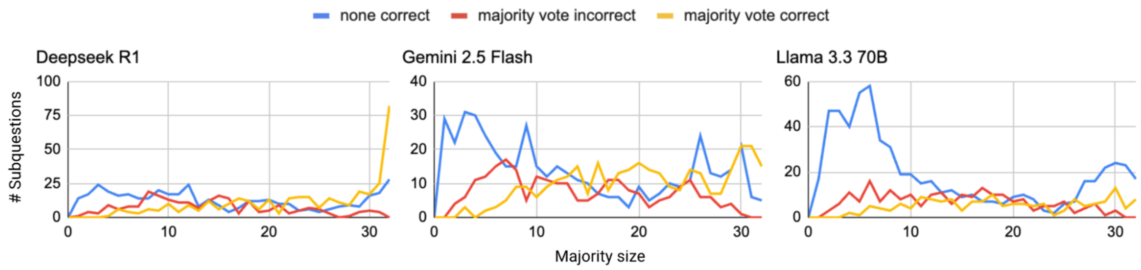

Figure 8: Distribution of the majority size when: no answers are correct; some answers are correct but not the majority vote; the majority vote is correct (sample size: 32)

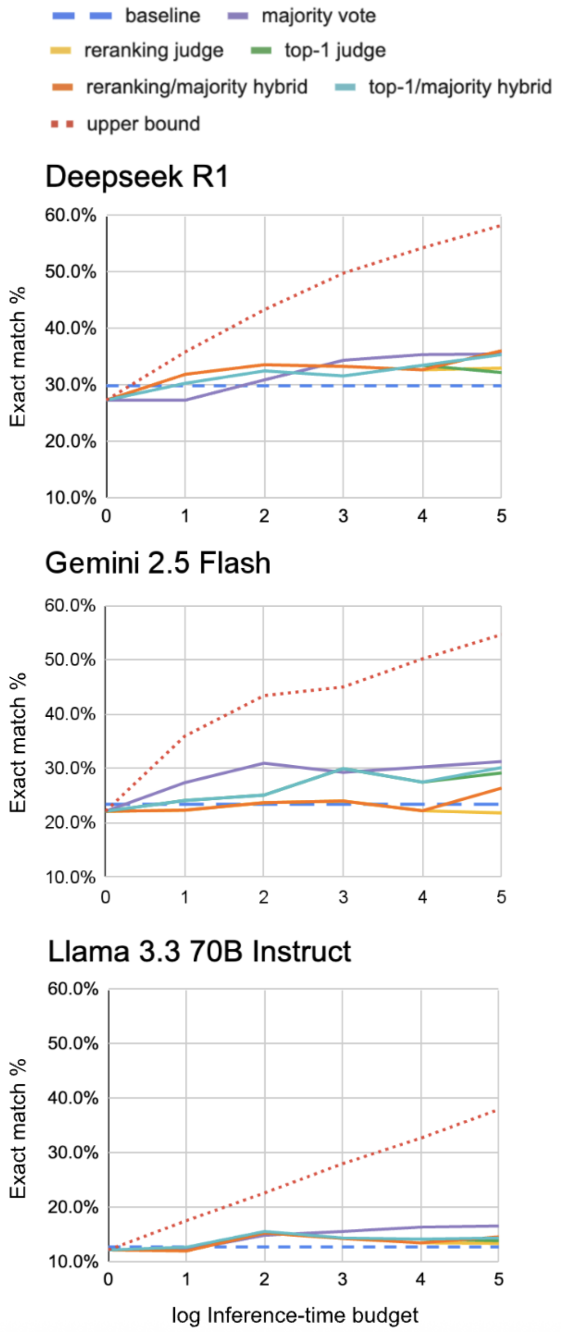

Figure 9: Comparison of aggregation methods and theoretical upper bound for each model

assign each response an apparent 'likeliness' score out of 1 instead (e.g. 0.8, 0.5, 0.1). To ensure a set of judged responses is obtained for all models, a set of scores between 0 and 1 with a distinct maximum is considered to be a valid judgement, yet even with the inclusion of these judgements, this method does not exceed the performance of the 'random judge' baseline.

Top-1 judge The top-1 judge approach, where the judge LLM is asked to select only the best answer, appears to reduce this problem difficulty and partially improve performance, offering a 7.3 percentage point improvement over the reranking judge for the Gemini model (32-sample); however, it performs similarly to the reranking judge for the other two models, and is consistently outperformed by the majority vote. Interestingly, the judge is found to be most accurate when judging between only two or four options, suggesting that a pairwise judging approach may improve performance; this is left for future work.

Judge/majority hybrid The majority size distribution (Figure 8) suggests that for 'doable' questions (where the model is correct at least once), the majority vote answer is correct 'more often than not' for the DeepSeek and Gemini models if this majority size exceeds approximately 16/32 samples (50%), and so the threshold for the hybrid approach is set at 50%: in other words, if there is a >50% majority, the majority answer is selected; otherwise, the LLM judge's answer is selected. This hybrid approach outperforms the pure LLM-as-judge method in all cases, but offers improvement above pure majority voting for the R1 model only, and this improvement is marginal.

Although these judging methods reduce the pool of possible answer options, we hypothesise why this may not have a positive effect on the models' performance. LLM-as-judge approaches are most effective for tasks where solving the problem, and verifying the correctness of the answer, are two clearly delineated tasks (Gu et al., 2024); yet in the linguistic puzzle domain, these two tasks are seen to be essentially inextricable - reasoning traces suggest that the 'judge LLM' generally must solve the problem on its own, and then select the answer closest to its solution - meaning the 'judge' offers little advantage over simply taking a sample of size 1.

4.3 Induction-application prompting with self-consistency

4.3.1 Summary

As shown in Figure 10, the induction-application method improves the baseline performance for all three models across the three applicable puzzle formats, with the marginal exception of the Llama 3.3 70B model on the Pattern format. This improvement is minor for the powerful DeepSeek R1 model on the Rosetta and Pattern puzzles, likely because this model often uses an induction-application approach to solve these problems without requiring specific prompting, yet for the numerical Monolingual puzzles, this approach offers a significant 18 percentage point baseline improvement. This suggests that only a format-specific algorithm, rather than an algorithm specific to each individual puzzle (as LINGOLY-TOO's small study examines), may be sufficient to improve model performance in this domain.

When self-consistency (with majority voting) is combined with this induction-application approach, the positive effect of both methods appears to be compounding. Performance generally improves as the simulated inference-time budget increases; however for the Pattern and Monolingual formats, the majority vote's accuracy appears to plateau for sample sizes of approximately 8 and above, suggesting that even medium-sized samples are sufficient to produce a diverse pool of outputs. This approach is particularly effective in the predominant format, the Rosetta puzzles, resulting in an improvement of 7.8, 13.8, and 2.8 percentage points for the R1, Gemini 2.5 Flash, and Llama 3.30 70B models respectively. This improvement is most significant in the Gemini model, outperforming even the much more 'deliberative' R1 for two of the three tested puzzle formats when this method is applied; this indicates that this speed-optimised model's poor baseline performance may reflect a

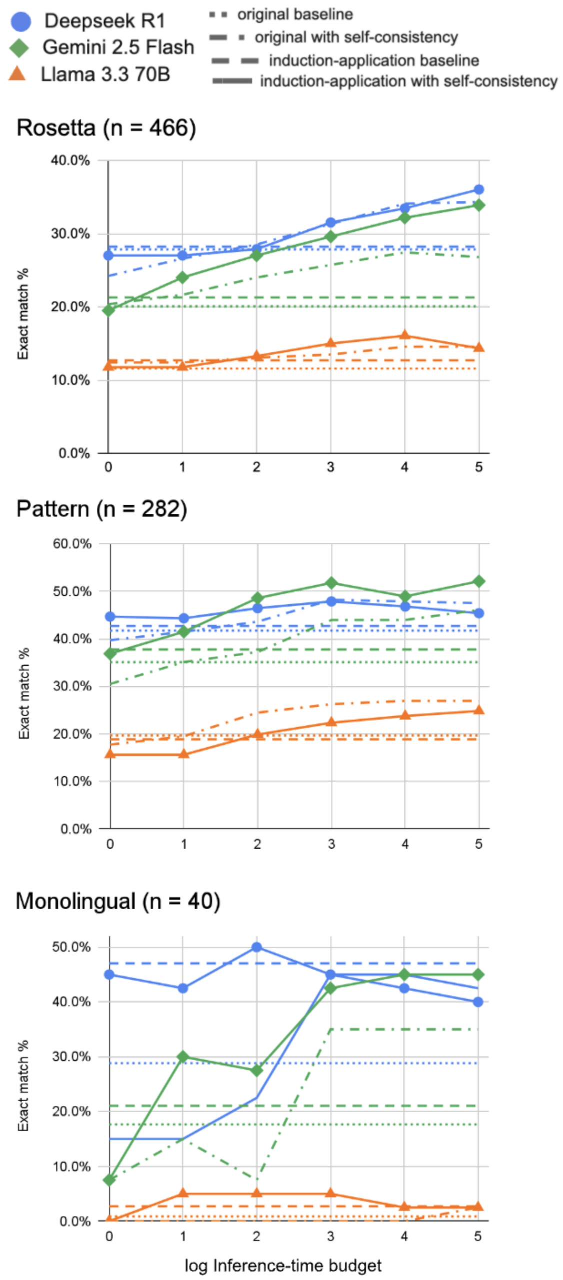

Figure 10: Effect of using induction-application prompting and self-consistency with increasing simulated inference-time budget for Rosetta, Pattern, and Monolingual formats

tendency to 'shortcut' to the output, rather than its 'true' reasoning capability in the linguistic puzzle domain, which can be unlocked by increasing inference-time budget and providing the model with algorithmic guidelines.

4.3.2 Measuring 'promising' thoughts

Using the sample-of-32 from each model for the three formats, logistic regression with mixed effects (controlling for the disparity in difficulty across subquestions) is performed to determine the effect size of answering the first subquestion in a problem sheet correctly, on the probability of answering the following subquestions correctly; the results are shown in Table 1.

From these results, we see that for the R1 and Gemini 2.5 Flash models, answering the first subquestion correctly in a given context window is a highly significant predictor that the model answers the following subquestions correctly; in other words, correct answers tend to come in 'clusters'. The anomalous result for the Gemini model on the Monolingual problems - where the model is 18.80 times more likely to answer subsequent subquestions correctly if it answers the first subquestion correctly - is partially due to there only being a small number of subquestions of this format (n = 40), and partially due to the nature of the format; once the solver successfully determines the meaning of each numeral, the questions themselves are often trivial. Accordingly, it was inferred that in general, when models take 'promising' reasoning pathways (as indicated by answering the first subquestion correctly), they are more likely to answer many of the subquestions correctly, but when they take 'poor' reasoning pathways they are likely to answer few of the subquestions correctly, suggesting that their outputs may follow 'logically' from their reasoning.

Conversely, this same factor is an insignificant or only somewhat significant predictor of correctness for the Llama model. This is likely partially due to this model's poor performance overall, leading to few correct answers and hence a biased sample, but also due to this model's higher output stochasticity; it may take a promising reasoning pathway, but then consequently produce outputs that do not follow from this, indicating that this model acts as an 'illogical' reasoner.

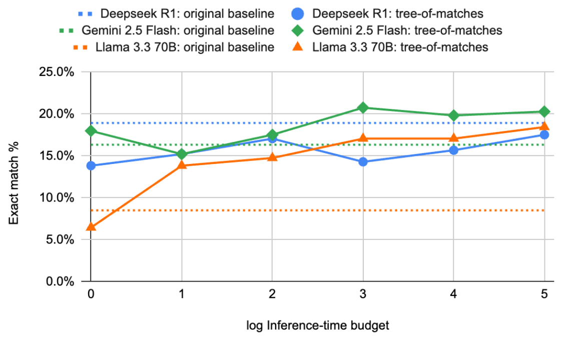

Figure 11: Effect of increasing inference-time budget on Tree-of-Matches performance (note that as this method involves fallback to the original prompting answer in the case of all responses being invalid, there is no sensible definition of a tree-of-matches 'baseline')

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 11. As with the other formats, performance generally improves with increasing inference-time budget, though these trajectories appear quite unstable, likely due to the high rate of invalid or unparsable responses (Table 2) leading to smaller 'actual' sample sizes. This invalid response rate is attributed partially to the intricate nature of these problems, but also to the sequential prompting framework used for this format being often too complicated for the models to follow correctly. To improve upon this method, it may be valuable to repeatedly prompt the model until a valid response is obtained, though this may increase the risk of infinite cycles or "context rot"; this is left for future work.

The R1 model in fact performs worse than the baseline when this sequential prompting method is used; this appears to be principally due to several subquestions which produce few parsable responses, leading to a very small sample pool from which to draw the 'most consistent' answer at each stage. To produce a more complete set of responses at each 'level' of the tree, it may be valuable to repeat this experiment with a larger number of samples.

However, for the other two models, using this approach improves performance on this puzzle format; this is especially true for the lightweight Llama model, whose exact match % increases by 9.9 percentage points over the baseline to 18.4%, almost on par with the much larger R1 model's performance. This is partially attributed to the

Table 1: Results of logistic regression with mixed effects (significance codes: *** $\rightarrow$ p<0.001, ** $\rightarrow$ p<0.01, * $\rightarrow$ p<0.05, ns = not significant)

Model	Puzzle Format	Odds Ratio	95% CI	P-value	Signif-icance
R1	Rosetta	1.60	[1.39, 1.86]	<0.0001	***
R1	Pattern	1.55	[1.32, 1.82]	<0.0001	***
R1	Monolingual	2.70	[1.57, 4.64]	0.0003	***
Gemini 2.5 Flash	Rosetta	2.29	[2.00, 2.62]	<0.0001	***
Gemini 2.5 Flash	Pattern	1.93	[1.66, 2.25]	<0.0001	***
Gemini 2.5 Flash	Monolingual	18.80	[11.54, 30.62]	<0.0001	***
Llama 3.3 70B	Rosetta	1.34	[1.07, 1.67]	0.011	*
Llama 3.3 70B	Pattern	1.30	[1.00, 1.68]	0.046	*
Llama 3.3 70B	Monolingual	1.92	[0.56, 6.56]	0.296	ns

Table 2: Rate of valid responses for the sequential prompting method (from the sample-of-32)

Model	% Subquestion responses valid	% Subquestions with $\geq$ 1/32 valid responses
R1	77.98%	88.02%
Gemini 2.5 Flash	50.89%	87.56%
Llama 3.3 70B	38.8%	72.81%

method reducing the frequency of instances where the lightweight model resorts to listing the matches in alphabetical order, forcing a more deliberate, step-by-step approach. However, inspection of individual questions suggests that while this method may improve the models' likelihood of correctly pairing the generally easier first few matches, it does not help the models correctly pair the later, more difficult matches - the true challenge of the problem - suggesting that there is still significant room for improvement in this format.

4.5 LINGOLY-TEAM: Overall Performance

By combining the results of applying self-consistent induction-application to the Rosetta, Pattern, and Monolingual formats, with the results of applying the Tree-of-Matches method to the Match-up puzzles, we obtain a set of results for all puzzles in the filtered benchmark, summarised in Figure 12 (full results for all methods are provided in §E). Overall, this combined LINGOLY-TEAM approach offers significant improvement over the baseline for all three models. For the R1 model, this method performs marginally worse than using 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, suggesting that this method may be best suited to boosting the performance of faster, mid-sized models. The Gemini model in particular improves markedly using this method to a score of 36.5%, an increase of 5.3 percentage points above the previous best approach (majority voting using the original prompting framework), and 13.1 points above its baseline performance, outperforming even the much more powerful R1 model's best score (36.0%).

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.

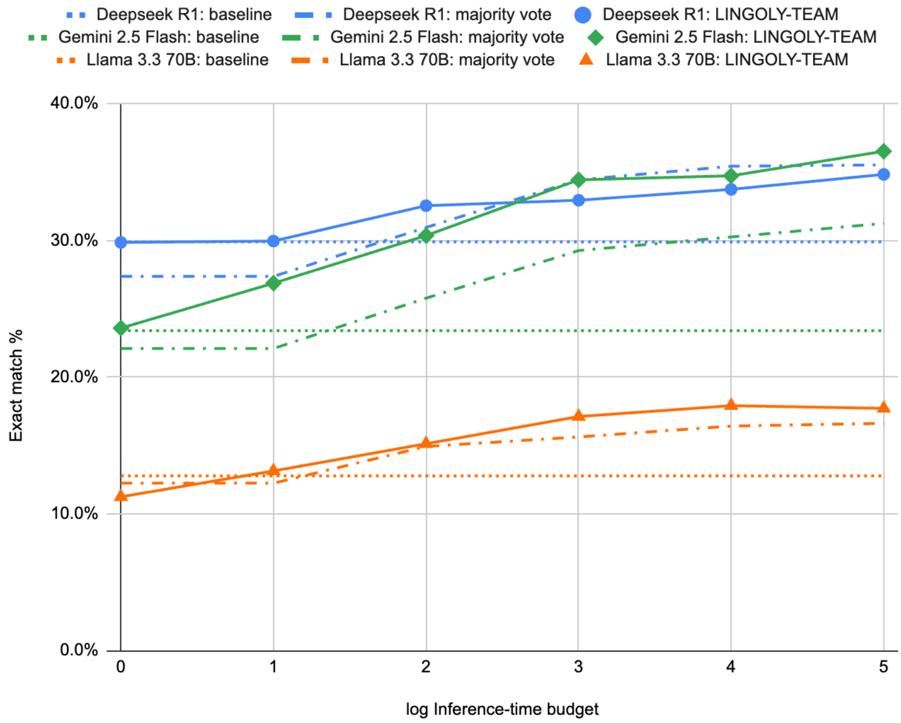

Figure 12: LINGOLY-TEAM's performance, compared with majority voting (using the baseline prompt) and the baseline

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., 2024a); 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'.

7 Ethics and data privacy

No human participants are involved in this project. The puzzles in Linguistics Olympiad competitions are either written by native speakers, or use languages whose speakers have consented to their publishing by linguists. The dataset also only includes puzzles whose authors have given permission for their work to be included in the LINGOLY-TOO benchmark (Bean et al., 2024; Khouja et al., 2025).

Although no personal data is handled in this project, the LINGOLY-TOO benchmark is dependent on its constituent dataset remaining 'unseen' to LLMs, and hence should be treated with care (Khouja et al., 2025). To preserve the integrity of the benchmark, we do not publish any of the obfuscated puzzles or solutions as examples in this paper, only publishing the original versions that are already publicly available. In our public codebase, datasets are placed in password-protected .zip files to prevent scraping by web crawlers.

Limitations

One obfuscation per problem: Unlike the original LINGOLY-TOO benchmark, which includes six obfuscations per problem, our dataset is filtered to only one obfuscation per problem. Although different obfuscations should theoretically be of equal difficulty, 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; due to resource constraints, 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 may not reflect the true inference-time budget when comparing across models, as some models may spend more time and/or tokens on reasoning for each sample. Of the models tested, the R1 model spends significantly more time reasoning for each problem, though the other two models generally output significantly more tokens, rendering the true inference-time budget 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.

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. However, these formats are generally distinct from each other and consistent across competition papers, rendering this task 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.

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.

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, and 1 others. 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.

D. Guo, D. Yang, H. Zhang, and 1 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. Nature, 645:633–638.

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. Preprint, arXiv:2507.15855.

IOL Committee. 2024. International linguistics olympiad rules and guidelines. `https://ioling.org/guidelines/en/`.

Jude Khouja, Karolina Korgul, Simi Hellsten, Lingyi Yang, Vlad Neacsu, Harry Mayne, Ryan Kearns, Andrew Bean, and Adam Mahdi. 2025. Lingolytoo: Disentangling reasoning from knowledge with templatised orthographic obfuscation. Preprint, arXiv:2503.02972.

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

Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024b. Retrieval augmented generation or long-context LLMs? a comprehensive study and hybrid approach. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 881–893, Miami, Florida, US. Association for Computational Linguistics.

Zheng-Lin 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), pages 262–269, Taipei City, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Neh Majmudar and Elena Filatova. 2025. Can llms solve and generate linguistic olympiad puzzles? Preprint, arXiv:2509.21820.

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.

Linlu Qiu, Liwei Jiang, Ximing Lu, Melanie Sclar, Valentina Pyatkin, Chandra Bhagavatula, Bailin Wang, Yoon Kim, Yejin Choi, Nouha Dziri, and Xiang Ren. 2024. Phenomenal yet puzzling: Testing inductive reasoning capabilities of language models with hypothesis refinement. In The Twelfth International Conference on Learning Representations.

Raghav Ramji and Keshav Ramji. 2024. Inductive linguistic reasoning with large language models. Preprint, arXiv:2412.17819.

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.

Inacio Vieira, Will Allred, Séamus Lankford, Sheila Castilho, and Andy Way. 2024. How much data is enough data? fine-tuning large language models for in-house translation: Performance evaluation across multiple dataset sizes. In Proceedings of the 16th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track), pages 236–249, Chicago, USA. Association for Machine Translation in the Americas.

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 Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. Preprint, arXiv:2201.11903.

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.

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 Appendix A: Example Puzzles

A.1 ‘Rosetta (Stone)’

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

...

Figure 13: ‘Rosetta Stone’ linguistic puzzle (source: United Kingdom Linguistics Olympiad 2013, Round 2, Question 3: Beja)

A.2 ‘Pattern’

Context:

1st person	**3rd person**	**Translation**
nàan̆	nòŏn̆	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**
lwòŏŏj	b)	to be different
d)	cèĕm	to eat
...		

Figure 14: ’Pattern’ linguistic puzzle (adapted from: United Kingdom Linguistics Olympiad 2022, Round 2, Question 4: Dinka)

A.3 ‘Monolingual’

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

...

Figure 15: ’Monolingual’ linguistic puzzle (source: United Kingdom Linguistics Olympiad 2019, Round 1, Question 8: Gumatj)

A.4 ‘Match-up’

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) based on matches identified:

Q 6.2 Translate:

(a) Ku kërceu?

...

Figure 16: ’Match-up’ linguistic puzzle (source: United Kingdom Linguistics Olympiad 2023, Round 1, Question 6: Albanian)

B Appendix B: 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.

Table 3: Manual corrections to answers in the dataset

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.

C Appendix C: 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 17.

We test this with the DeepSeek R1 model only, perform self-consistency on subsamples of size 16, 8, 4, 2, and 1. This performance is compared to the ‘original’ baseline prompt in Figure 18.

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 17: Baseline prompt sketch, from the LINGOLY-TOO benchmark

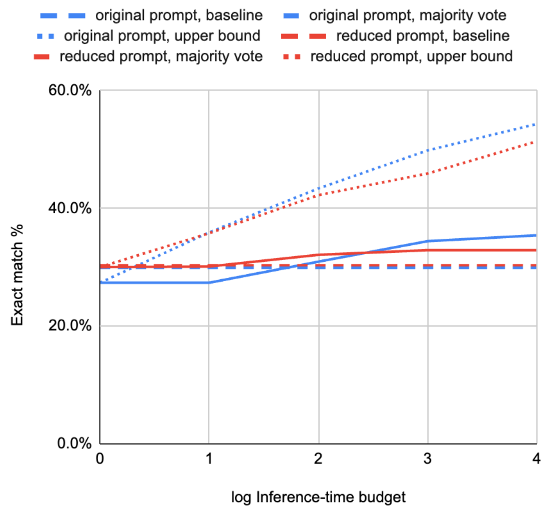

Figure 18: Comparison of original and reduced prompt performance for all puzzle formats (using the DeepSeek R1 model)

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 was abandoned for the remainder of this project, in favour of the original baseline prompt from the LINGOLY-TOO benchmark.

D Appendix D: LINGOLY-TEAM Prompts

D.1 ‘Rosetta (Stone)’

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 19: Induction-application prompt sketch for a ‘Rosetta’ puzzle

D.2 ‘Pattern’

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

D.3 ‘Monolingual’

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

D.4 ‘Match-up’

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? a) Did you drink anything?
2. Kujt ia shiti? b) Did you kill someone?
3. Kë vrau? 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 22: Induction-application prompt sketch for a ’Match-up’ linguistic puzzle (source: United Kingdom Linguistics Olympiad 2023, Round 1, Question 6: Albanian)

E Appendix E: Full Results

E.1 DeepSeek R1

Table 4: Comparison of methods for the DeepSeek R1 model (baseline: 29.9)

Sample size	Majority vote	Upper bound	Rerank judge	Top-1 judge	Rerank/ majority hybrid	Top-1/ majority hybrid	LINGOLY-TEAM
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

E.2 Gemini 2.5 Flash

Table 5: Comparison of methods for the Gemini 2.5 Flash model (baseline: 23.4)

Sample size	Majority vote	Upper bound	Rerank judge	Top-1 judge	Rerank/ majority hybrid	Top-1/ majority hybrid	LINGOLY-TEAM
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

E.3 Llama 3.3 70B Instruct

Table 6: Comparison of methods for the Llama 3.3 70B Instruct model (baseline: 12.8)

Sample size	Majority vote	Upper bound	Rerank judge	Top-1 judge	Rerank/ majority hybrid	Top-1/ majority hybrid	LINGOLY-TEAM
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