LOBSTER🦞: Linguistics Olympiad Benchmark for Structured Evaluation on Reasoning

Da-Chen Lian
Graduate Institute of Linguistics
d08944019@ntu.edu.tw

Ri-Sheng Huang
Dept. of CSIE
r13922102@csie.ntu.edu.tw

Pin-Er Chen
Graduate Institute of Linguistics
f10142001@ntu.edu.tw

Chunki Lim
Graduate Institute of Linguistics
r14142001@ntu.edu.tw

You-Kuan Lin
Dept. of Elec. Engineering
conlangtaiwan@gmail.com

Guan-Yu Tseng
Graduate Institute of Linguistics
r14142007@ntu.edu.tw

Zhen-Yu Lin
Dept. of FLL
a0985026048@gmail.com

Pin-Cheng Chen
Dept. of FLL
b10102102@ntu.edu.tw

Shu-Kai Hsieh
Graduate Institute of Linguistics
shukaihsieh@ntu.edu.tw

National Taiwan University

Abstract

We propose the Linguistics Olympiad Benchmark for Structured Evaluation on Reasoning, or LOBSTER🦞, a linguistically-informed benchmark designed to evaluate large language models (LLMs) on complex linguistic puzzles of the International Linguistics Olympiad (IOL). Unlike prior benchmarks that focus solely on final answer accuracy, our benchmark provides concrete evaluation protocols and rich typological metadata across over 90 low-resource and cross-cultural languages alongside the puzzles. Through systematic evaluations of state-of-the-art models on multilingual abilities, we demonstrate that LLMs struggle with low-resource languages, underscoring the need for such a benchmark. Experiments with various models on our benchmark showed that IOL problems remain a challenging task for reasoning models, though there are ways to enhance the performance—for example, iterative reasoning outperforms single-pass approaches in both final answers and explanations. Our benchmark offers a comprehensive foundation for advancing linguistically grounded, culturally informed, and cognitively plausible reasoning in LLMs. [1]

Keywords: reasoning, large language model, benchmark, linguistics olympiad

1 Introduction

While advances in LLM have revolutionized natural language processing, significant challenges persist in achieving robust reasoning capabilities—particularly for tasks requiring multi-step abstraction, symbolic verification, and constraint-based hypothesis testing. Several reasoning-enhancement paradigms have emerged with the hope to solve more complex problems, such as hybrid tool-integrated approaches (He et al., 2025; Gao et al., 2025; Paranjape et al., 2023; Schick et al., 2023; Wu et al., 2025), or agentic systems (Li et al., 2025; Ke et al., 2025).

The International Linguistics Olympiad (hereinafter abbreviated as IOL; 2003-2025) presents uniquely challenging problems that require solvers to induce linguistic rules from micro-data, often in low-resource or unfamiliar languages. These problems test not just surface-level pattern recognition, but demand multi-step abstraction, structural reasoning, and cultural inference. Comprising four parts (see Appendix A.1), an IOL problem is meticulously crafted to be self-contained, without the necessity of any prior knowledge in linguistic rules. The logical consistency and sufficiency thus allows participants to decode the underlying linguistic rules purely through reasoning and pattern analysis (Bozhanov and Derzhanski, 2013), the low-resource nature of the languages in which these problems made offers an isolated envi-

[1] The benchmark and the source code can be found at https://github.com/lopentu/LOBSTER.

ronment to test the reasoning performance of models. (See Section 3)

In addition to abstract linguistic reasoning, some IOL problems incorporate elements that go beyond standard textual input, requiring models to process non-standard scripts, phonetic transcriptions, or visual symbol systems such as maps or family trees. Some problems involve rare or extinct writing systems—occasionally ones not yet fully encoded in Unicode—demanding the recognition and manipulation of unfamiliar glyphs (Shih et al., 2025). Others rely on International Phonetic Alphabet (IPA) representations, tone contour symbols, or constructed orthographies that encode morphophonemic information. A subset of tasks also includes pictographic cues, spatial arrangements, or logical diagrams (see Appendix A.2), which are essential to its decipherment. While recent vision-language models have made progress in visual and text input jointly, their ability to integrate these modalities with complex reasoning remains limited.

Another distinctive aspect of IOL problems lies in their **cross-cultural and semantic depth**. Beyond the structural reasoning over phonology, morphology, and syntax, many problems explicitly involve semantic inference, cultural conceptualization, or sociolinguistic reasoning—for instance, deciphering kinship terms, numeral systems, metaphorical extensions, or culturally situated deixes. These tasks compel both human and AI solvers to imagine how meaning might be constructed in unfamiliar cultural worlds, often requiring *cross-linguistic abstraction* or *anthropological imagination*. For LLMs, this poses a profound challenge: it tests their ability to generalize across not only linguistic structures but also cognitive and cultural domains. IOL problems, therefore, serve not only as puzzles of language form but as tests of situated meaning-making and cultural flexibility, offering a rigorous probe into the limits of LLMs' representational and interpretive capacity across diverse human experiences.

These complex challenges expose the limitations of current LLMs and existing evaluation methods, which often prioritize final-answer accuracy over the reasoning process.

2 Review of Past Studies

Reasoning models and reason-enhancing paradigms enable LLMs to actively explore solutions, rather than just passively generate text. Their efficiency is frequently evaluated through human-level reasoning benchmarks like the International Linguistics Olympiad (IOL) (Şahin et al., 2020; Chi et al., 2024), where success requires inferring linguistic structures from constrained datasets, mirroring real-world challenges in rule abstraction, cross-linguistic generalization, and constraint satisfaction.

2.1 Reasoning on Linguistic Structures

Reasoning on linguistic structures presents unique challenges, when compared to other reasoning domains such as math or coding. Unlike purely symbolic systems, understanding human languages requires world knowledge, cultural context, and common sense. For example, the word for "five" and "hand" is the same in some languages because there are five fingers on a hand. This requires the model to also infer of a semantical link between the two senses; it is inconceivable from a symbolic inductive logical perspective.

For the classic Rosetta Stone problems,[2] the inference task is in a sense a more complex variant of the "infer one form of a word/phrase/sentence to another" task.

The induction task has long been of interest to linguists (Durham and Rogers, 1969), as it mirrors what linguists do in a field study. This induction task has been framed in at least two ways. One perspective treats it as a program synthesis problem, where the goal is to generate a "program"—a set of formal rules—that transforms inputs to outputs (Naik et al., 2024). This has led to the development of domain-specific languages for expressing such string transformations (Vaduguru et al., 2021). Alternatively, the task can be viewed as constrained text generation, where specialized architectures are designed to model linguistic phenomena (Lu et al., 2024).

A complementary line of research explores augmenting LLMs with explicit linguistic knowledge. Rather than relying solely on induction from examples, this approach provides models with resources like dictionaries, morphological analyzers, or grammar books, mimicking how a human linguist might consult reference materials (Zhang et al., 2024). While the ability to leverage such

[2] Given a set of sentences in an unknown language and their corresponding translations, the agent should infer the underlying rules, such as grammar, meaning of each word, or spelling changes in the unknown language.

grammatical descriptions can be systematically evaluated (Tanzer et al., 2024), their utility is task-dependent: for translation, performance gains stem from parallel examples rather than grammatical explanations, which are better suited for targeted linguistic analysis tasks (Aycock et al., 2025). Such nuances call for more research on the intersection of LLMs and linguistics expertise.

2.2 Relevant Benchmarks from Linguistics Olympiads

To evaluate the capabilities of LLMs on complex reasoning tasks, researchers have developed various benchmarks. The following are some benchmarks relevant to Linguistics Olympiad problems:

- **LingOly** (Bean et al., 2024):[3] With 1,133 linguistic puzzles from the UK Linguistics Olympiad (UKLO),[4] it excludes image-based puzzles, non-Latin scripts, and open-ended questions to ensure machine-scorability. The evaluation is exact-matched, excluding fuzzy matches and normalizing Unicode variations, to ensure linguistic precision. Less strict metrics like ROUGE and BLEU were analyzed, but the primary focus remains on context-dependent reasoning.

- **Linguini** (Sánchez et al., 2024):[5] This benchmark also extracted data from IOL problems, covering low-resource languages and three core task types: sequence transduction , fill-in-the-blanks, and number transliteration (i.e. digit-to-text conversion). The evaluation uses exact match accuracy and the softer chrF metric to assess performance on structured linguistic inference.

- **IOLBENCH** (Goyal and Dan, 2025):[6] 90 of the IOL Problems were digitalized into text or structured representation through LLM-based document recognition, including some multimodal components. While it take cares of free-response answers through different grading metrics, the LLM-based unverified data construction made most of the problem in the dataset ill-formed.

Existing benchmarks for IOL-style tasks have demonstrated the promising capabilities of LLMs in handling complex linguistic reasoning. However, several critical limitations remain that constrain both fine-grained evaluation and meaningful model improvement.

First, most current evaluations rely predominantly on exact-match accuracy of the final answers, without considering the plausibility, internal consistency, rules used to explain the answers, or are logical coherence of intermediate reasoning steps. This narrow focus obscures whether models are genuinely applying linguistic principles or merely relying on pattern recognition and heuristic guessing. Such a limitation hampers our ability to diagnose reasoning failures and systematically improve model understanding.

Specifically, these methods often (i) lack rigorous alignment with linguistic knowledge bases, (ii) fail to capture the reflective, iterative, and self-corrective nature of human linguistic reasoning, and (iii) inadequately represent the hierarchical and multi-layered reasoning structures characteristic of IOL challenges. As a result, existing evaluation paradigms are insufficient for capturing the depth, correctness, and explanatory richness of linguistic problem-solving processes. This highlights the need for more sophisticated evaluation methodologies specifically tailored for linguistic reasoning contexts.

3 Motivation: Probing the Limits of LLMs

As Joshi et al. (2020) highlight, the vast majority of the world's languages are low-resource, and their unique linguistic features are underrepresented in pre-training corpora. This skew towards high-resource languages like English hinders model performance and the potential for cross-lingual transfer, even for typologically similar languages (Pires et al., 2019).

To empirically ground the need for a more nuanced evaluation benchmark, we assessed a state-of-the-art model, Gemini-2.5-flash, on a multilingual translation task using the FLORES-200 dataset (NLLB Team et al., 2022). Our experiment, which covered 204 languages, revealed critical limitations (see Appendix I for full details). We found that:

1. Performance is strongly correlated with resource availability. The model frequently

[3] Relevant resources for LingOly can be found on GitHub: https://github.com/am-bean/lingOly.

[4] https://www.uklo.org/

[5] Relevant resources for Linguini can be found on GitHub: https://github.com/facebookresearch/linguini

[6] Relevant resources for IOLBENCH can be found on GitHub: https://github.com/Satgoy152/ling_llm

failed to generate any output for the lowest-resource languages (Class 0).

2. A significant performance asymmetry exists based on translation direction. The model performed substantially worse when translating from English *to* a target language ($E \rightarrow T$) than in the reverse direction ($T \rightarrow E$), especially for low-resource languages.

3. Statistical analysis confirmed that language family and resource class are highly significant predictors of translation quality, while script was not.

These findings demonstrate that even powerful models struggle with genuine multilingual tasks, often failing at the basic level of text generation for a large portion of the world's languages. This underscores the inadequacy of benchmarks that focus only on high-resource languages or overlook reasoning failures, motivating our development of LOBSTER🦞.

4 LOBSTER🦞: Linguistics Olympiad Benchmark for Structured Evaluation on Reasoning

The IOL problems exhibits a wide range of typological diversity, an essential step in understanding the nature of such a benchmark in profiling the distribution of languages, for which existing LLM benchmarks rarely account. Regarding language family, the most common language families are North American, Austronesian, Indo-European, and African (see Appendix D for the language family distribution). However, There remains a gap in understanding how models perform across different language families and typological features.

LOBSTER🦞 is built on a curated selection of past IOL problems. Unlike prior datasets, it includes enriched metadata that allows for deeper linguistic diagnostics and reasoning trace comparison. Our benchmark is intended to support: (i) accurate transcription of contents of IOL problems; (ii) typologically grounded performance analysis; and (iii) assessment of models' cross-cultural and cross-linguistic inference abilities.

4.1 Data Construction

Our benchmark consists of 96 problems (225 subproblems) sourced from the IOL archive (2003–2024). For kinship problems[7] involving family trees, we convert the graphical representations into textual relationship descriptions (see Appendix A.3 for an example of a kinship problem). We exclude problems that fully rely on image-based information or untranscribable symbols.

Since most IOL problems provide only the final solutions along with some grammatical rules, without including detailed reasoning steps, we use Gemini-2.5-pro to generate structured step-by-step solutions as gold-standard references in the benchmark. The LLM is prompted to act as a linguistics expert, producing logical deductions, linguistic rules, and problem-solving strategies that lead to the official solutions (see Appendix B for the prompt template). To ensure reliability, seven human experts and three IOL contestants manually verify and refine these reasoning chains, resolving any inconsistencies to ensure alignment with the official IOL solutions.

In summary, for each IOL problem in our benchmark, we include the transcribed problem text, the official solution, and the expert-verified, refined, LLM-generated reasoning. The latter is not used for grading but serves as a qualitative reference for human-understandable reasoning processes.

4.2 Typological Annotation

In addition, each problem within LOBSTER🦞 is annotated along multiple linguistic dimensions to facilitate a structured analysis of model performance. The current typological and problem-oriented schema is an adaptation of the UKLO classification framework[8] with the annotation being carried out by seven linguistic experts. We annotate three categories for each problem: Subject, Type, and Theme; the respective tags are detailed below, while the descriptions of each tag are shown in Appendix C.1. Also, the Glottocode is included (Hammarström et al., 2024) for each problem. Table 2 shows an example of annotations for one problem.

The distribution charts of each typological category in our benchmark are shown in Appendix D. Key findings include:

Subject and Type Distribution: Referring to Appendix E, the data suggests that Syntax and

[7] Kinship problems focus on understanding how different languages and cultures describe family relationships and naming systems.

[8] https://www.uklo.org/technical-information/

Morphology are the most prominent subjects in IOL problems, with Rosetta type problems being heavily focused in these areas (i.e., 17.4% and 16.5%). Semantics are distributed across multiple problem types (0.9%, 6.4%, 3.7%, 0.9%, 7.3%) compared to others. Overall, the uneven distribution implies that certin problem types are strongly associated with particular subjects (e.g., Phonology has a spike (13.8%) in Pattern type problems), while others are more diffuse.

Subject and Language Family Distribution: North American languages have the highest number of problems (14), followed by Austronesian (11), Indo-European (10), and African (10). As shown in Appendix F, Syntax is the most widely represented subject, appearing in 7 out of the top 10 language families, with the highest concentration (6.2%) in African. Morphology is the second most frequent, appearing in 9 out of the top 10 families, with multiple mid-range values (2.5%–5.0%). While Phonology stands out in Indo-European and North American, Semantics is more broadly distributed, with Austronesian, African, Australian, and Niger-Congo all having moderate percentages (around 2.5%). In summary, Syntax, Morphology, and Phonology dominate the subject distribution, with North American, Austronesian, Indo-European, and African languages showing the richest variety of subjects. More details are shown in Figures (a) and (d) in Appendix D.

Type and Language Family Distribution: Regarding Appendix G, *Match-up* problems are more common in Austronesian and North American language families. *Pattern* problems are particularly prevalent in Indo-European languages. *Rosetta* problems are the most common overall (44 problems), appearing across various language families, with especially high occurrences in African and North American languages. More details are shown in Figure 8 (b) and (d) in Appendix D.

These findings reinforce the relevance of typological and reasoning-aware annotations. They also highlight the inadequacy of answer-only metrics in capturing the richness of linguistic cognition demanded by IOL problems.

4.3 Evaluation Protocol and Metrics

Existing IOL-styled benchmarks (Bean et al., 2024; Sánchez et al., 2024; Goyal and Dan, 2025) tend to rely on exact string matching for accuracy, which fails to award partial credit for complex problems. Grading IOL solutions is rather complex and flexible. Generally, the final answer is not the sole contributor to the final score; the explanation of grammatical rules is just as important. We hence evaluate the final solution generated by the model with respect to the rules provided in official solutions.

4.3.1 Evaluation of the Final Solution

First, we assess the model-generated final solution based on two distinct components: the **answer** and the **explanation of rules**.

The **answer** refers to all the questions inside the problem, which the contestant would be asked to answer. For example, the sample problem in Appendix A.1 contains 9 questions (1 in subproblem (a), 3 in (b), and 5 in (c)). Most of the questions, such as short sentence translations, can be graded with simple string matching, but an exact match metric would be unsatisfactory in many cases. Examples include semantics problems where any synonym should be counted as correct if the term is inferred, but not copied from the problem; or questions that ask for an explanation to a certain linguistic phenomenon (not to be confused with the "explanation" part of the solution below). In these cases, various metrics can be applied (e.g., BLEU, sentence embedding) depending on the preferences of the user of our benchmark.

On the other hand, the **explanation** requires the model to write down the linguistic rules it inferred from the problem data. The official IOL problem sheet explicitly states, "Your answers must be well-supported by argument. Even a perfectly correct answer will be given a low score unless accompanied by an explanation", but the official grading rubrics are not publicly available, thus evaluating the quality of these free-text explanations poses a significant challenge unaddressed by past works. We address it with a two-stage procedure: Through **rule composition**, we convert the official solution into a discrete set of key linguistic rules, creating a gold-standard "rule checklist." We then employ an LLM grader, specifically Gemini-2.5-flash-lite, in the process for **checklist grading**. The grader is prompted to compare the model's generated explanation against our rule checklist and determine the number of gold-standard rules that were correctly described. By grading with a checklist rather than the official, free-form solution, we reduce subjectivity in the grading criteria, and minimize potential biases (e.g., self-preference) from the LLM grader.

This approach enables a stable, fine-grained, and quantitative assessment of the explanation's quality. The total score for the final solution is a weighted combination of the scores of "answer" and "explanation of rules." By default, we assign equal weight (50/50) to each component, with points distributed evenly across all subproblems for the answer and all identified rules for the explanation. With additional scores granted to the explanation, the benchmark we propose can show whether the model answers through reasoning within the problem data or through other external confounders.

5 Testing LOBSTER🦞 on Different Systems

In the previous sections, the multilingual abilities of LLMs are shown to be inadequate. Therefore, when attempting to solve an IOL problem, LLMs may not solely rely on prior knowledge about the target language or typology. To pinpoint the ability of state-of-the-art models on IOL problems, we examined a range of models on LOBSTER🦞, and verified that IOL problems pose a challenge even for state-of-the-art reasoning models.

5.1 Setup

A set of experiments was conducted using the most powerful models within budget. In addition to directly prompting the models, we also tested with various settings for the same model. To ensure numerical stability, for each problem in each setting, we obtained 5 samples and averaged over the scores. The settings include:

- **Vanilla baseline**: A direct, single-pass call to an LLM to solve the problem, following the required output format. We used OpenAI-o4-mini, Gemini-2.5-pro, and GPT-5 for the experiments, with temperature set to 0.75.

- **Guided prompt**: A major drawback of the vanilla prompting is that, usually the LLM is not familiar with the underlying assumptions of Linguistics Puzzles (e.g., "All the questions are self-contained", "The final solution should be able to explain 100% of the examples, not just 90%"). To inform the model about such nuances, we include the Introduction chapter of the book *Linguistics Olympiad: Training guide* (Neacşu, 2024) in the system prompt. As an introductory text about linguistics problem, the chapter describes the format and classification of a linguistics problem, guidelines for solution writing, and some toy examples.

- **Grammar agent**: Past work has shown that the model performs better when given explicit knowledge (Tanzer et al., 2024). In this setting, the model was provided with a reference grammar book of the target language. To do so, we constructed a database containing reference grammar books from publicly available resources, and manually labeled the language, with its Glottocode as metadata to facilitate search.

- **Mixture-of-Agents**: Following Mixture-of-Agents (MoA) (Wang et al., 2025), a multi-round framework is used, as depicted in Figure 1. The system consists of a customizable number of *Solver Agents* and *Aggregator Agents*. The idea is that iteratively collecting multiple proposed solutions may improve performance. In our setup, we used 2 agents for each layer (N=2 following the notations in the figure)—Gemini-2.5-pro and OpenAI-o4-mini. The solutions are iterated for at most 6 rounds (i.e., M=2, R $\in \{0, 1, 2, 3, 4\}$), with the last round being the "final aggregator" in the figure.

- **Single agent, multi-rounds**: Equivalent to the Mixture-of-Agent setting with N=M=1, the solution of a solver is fed into itself for multiple rounds. This setting disentangles the effect of parallel generation from iterative refinement.

5.2 Results and Analysis

5.2.1 Comparison between Models

The results are summarized in Figure 2. Based on the evaluation methods detailed in Section 4.3.1, the answer and the explanation scores are calculated separately, and a combined score ("total score") is also provided. An overview shows that the scores for the "answer" and the "explanation" are positively correlated (r=0.501). (See Appendix O)

Regarding the base models, our experiments are mainly comparing models based on Gemini-2.5-

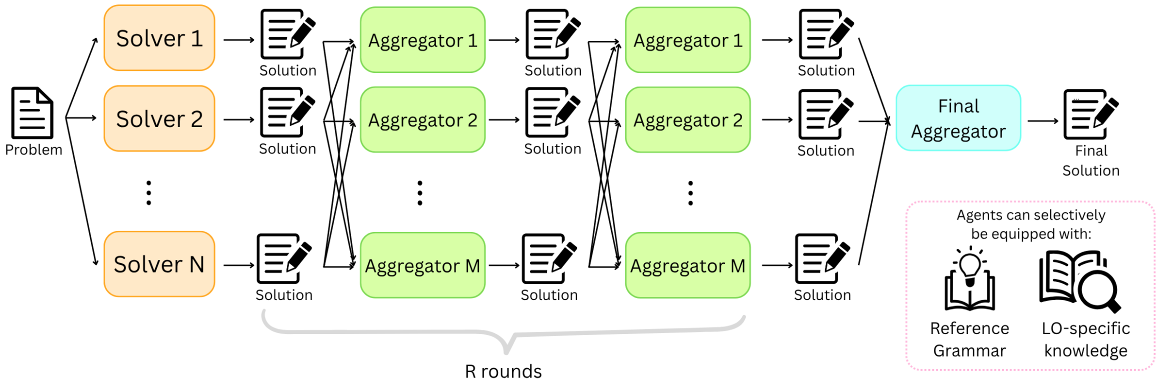


Figure 1: Multi-Agent Framework for Solving Linguistics Olympiad Problems

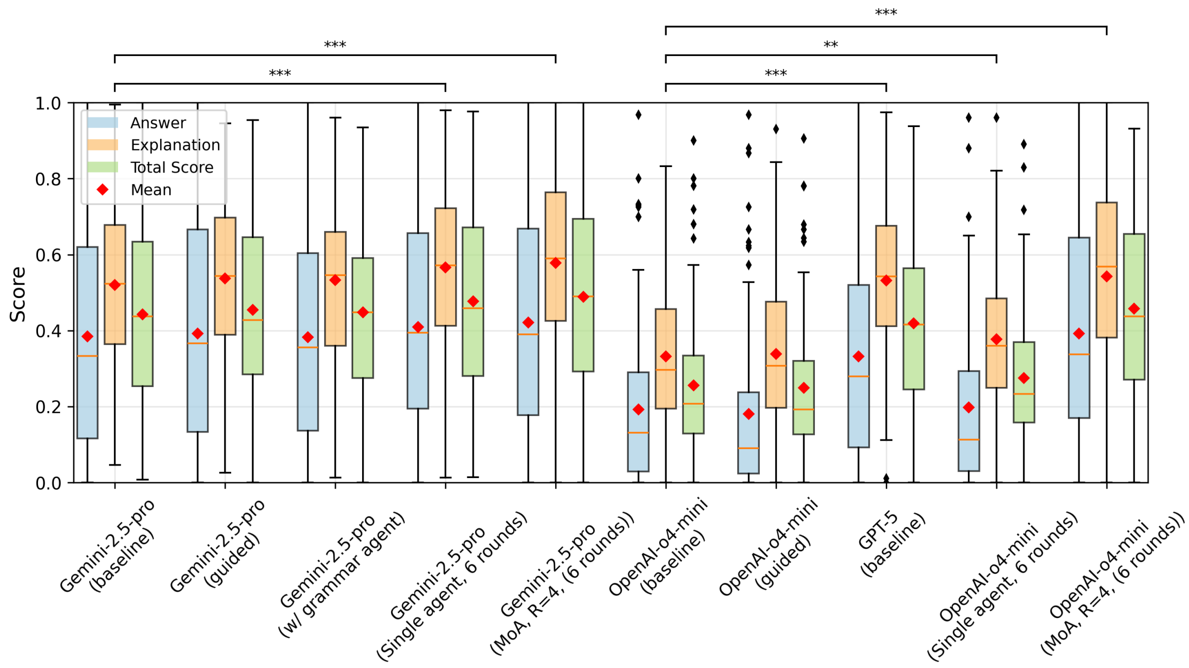


Figure 2: Scores on LOBSTER🦞of Different Models. Statistical significance was examined using paired Student's t-test. For simplicity, we only plot the significance between baseline vs. other models. {*,**,***} denotes $p < \{0.05, 0.005, 0.0005\}$, respectively. The model name in MoA denotes the final aggregator and R is the number of intermediate rounds.

pro and OpenAI-o4-mini. The former considerably outperforms the latter, and is marginally better than GPT-5.

The trends between different settings are less clear: we found no statistically significant difference in the grammar agents scores compared to the baseline, nor in guided prompts vs. baseline. These results contradicts our expectation of an improvement; for discussions on possible reasons, see Section 5.3.

On the other hand, Mixture-of-Agents gives steadily increasing scores as the number of rounds increases, which are significantly better ($p < 0.05$) than the baseline as long as there is more than one round. Interestingly, the final aggregator plays an important role in the performance—if the final aggregator is weak (in this case, OpenAI-o4-mini), even though it has seen the (better) solutions generated by other models (in this case, Gemini-2.5-pro), the output scores far lower than the stronger model.

A natural question arises as to whether the effectiveness of MoA comes from multi-round from multi-agent. We introduced the single-agent multi-round setting to isolate their effects. Results show that additional rounds consistently improve performance, confirming the benefit of iterative reasoning. The multi-agent effect, however, is less pronounced for Gemini-2.5-pro—likely because it is already a stronger model, and a weaker collaborator offers limited help ($p = 0.105$ for 6-round MoA vs. single-agent multi-round with Gemini-2.5-pro). In contrast, OpenAI-o4-mini benefits greatly when paired with Gemini-2.5-pro ($p < 0.0001$).

The exact scores can be found in Appendix M.

5.2.2 Performance regarding Language Family and Problem Type

To gain a more nuanced breakdown of the model's performance, we analyzed the Gemini-2.5-Pro statistics by categorizing the problems based on language family, linguistics subject (e.g., phonology, syntax), and problem type (e.g., *Pattern*, *Match-up*). The detailed scores are plotted in the Appendix N.

Typologically, the model performs best on language isolates (mean = 0.70), Turkic (0.64), and Indo-European (0.55) languages, but struggles with Papuan (0.29), South American (0.25), and Australian (0.34) ones. The trend may be partially attributed to the resource-level of the languages.

By problem type, the model achieves its highest scores on *Monolingual* problems and lowest on *Match-up*. Across linguistic domains, it performs worst on syntax and best on semantics. The "Others" category has a score surpassing all others, possibly due to intrinsic differences in problem design.

Overall, the model shows strong performance in certain areas but inconsistent reasoning across languages, subjects, and problem types.

5.3 Discussions and Limitations

Exposure to the target language during pretraining. Even though the languages are low-resourced, models may still have some prior exposure that gives them an advantage in problem solving, meaning scores may not reflect pure reasoning ability. Additionally, the Internet presence of IOL problems increased the possibility of being in the pretraining data for some models. One approach to mitigate this is to systematically adjust the orthography, making it harder for models to recognize the language while preserving the problem's content (Khouja et al., 2025). Our work provides a solid foundation well-suited for future use.

Unimodality. Currently, the benchmark is designed to handle only text, in order to be applicable to a wider range of models. However, linguistics problems may involve other modalities (e.g., visual data), as seen in problems involving writing systems, kinship trees, and even maps. Such problems could be transcribed into text if possible but are usually excluded from the benchmark.

The exact content of the Grammar Agent. Contrary to our expectation, we found no major improvement when a model was equipped with a reference book. Dissecting the reason for this observation is a non-trivial task because the content and format of reference grammar books vary greatly, creating many confounding variables. For example, as Aycock et al. (2025) have discovered, the example sentences may be more useful than long paragraphs of grammar descriptions.

Another possible reason lies in the complexity of language itself. Reference grammar books are not a unified or accurate reflection of language but rather artifacts that attempt to summarize the real-world language use. Consequently, for the same language, it is not uncommon for different sources to have different orthographical conventions for transcription, variations from the data (e.g., speaker/dialect variations), and conflicting theories about grammar, where later works may disagree with the past literature. In Tanzer et al. (2024), these inconsistencies did not emerge, and we hypothesize that this is because their work used the same, consistent source for benchmarking and knowledge provision.

In any case, investigating the nature of external knowledge is necessary to continue the study. Such studies may require high-quality classification and annotation of books broken down into meaningful units, which we anticipate will demand considerable manual effort.

Reasoning traces. While our benchmark is a leap forward from previous linguistic reasoning benchmarks (in particular, ours is able to evaluate partially correct solutions meticulously, and is rich in metadata), the "thought process" of a model is not taken into consideration when grading. To our knowledge, evaluating the reasoning steps of LLMs remains an open problem.

To help advance this line of research, we provide a dataset of the gold-standard reasoning traces alongside the quantitative grading part of the benchmark, and ensure that their formats are fully compatible. One possible quantitative use of the reasoning trace data is as a "rule checklist," similar to the explanation grading in Section 4.3. This dataset, for which direct applications are yet to be explored, invites future researchers interested in reasoning and human cognition.

6 Conclusion

In this work, we introduced LOBSTER🦞, a linguistically-informed benchmark designed to move beyond final-answer accuracy and enable a granular assessment of an LLM's reasoning on complex linguistic structures. Our typological analysis of IOL problems provides a structured lens for this evaluation, while our empirical study of a state-of-the-art model on the FLORES-200 dataset underscored the critical need for improved cross-linguistic generalization, particularly in low-resource settings. We call on the community to build on this foundation to look inward at the nascent logic of LLMs, and outward at the boundless diversity of language that inspires them.

Acknowledgments

We would like to thank Zi-Cheng Yang for his help with implementation, Tung-Le Pan (Chair of Jury of IOL 2025) for comments about the IOL, and the members of Taiwan Olympiad in Linguistics—especially Hung-Chi Chen, Yin-Shuo Chang, Kanoa Ziyang Teng and Chloe Cheng—for the annotation and refinement of the data. We would also like to thank the reviewers for their valuable feedback.

References

Seth Aycock, David Stap, Di Wu, Christof Monz, and Khalil Sima'an. 2025. Can LLMs really learn to translate a low-resource language from one grammar book? In *The Thirteenth International Conference on Learning Representations*.

Andrew Bean, Simi Hellsten, Harry Mayne, Jabez Magomere, Ethan A., Ryan Chi, Scott A. Hale, and Hannah Rose Kirk. 2024. Lingoly: A benchmark of olympiad-level linguistic reasoning puzzles in low resource and extinct languages. In *Advances in Neural Information Processing Systems*, volume 37, pages 26224–26237. Curran Associates, Inc.

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.

Stanton P. Durham and David Ellis Rogers. 1969. An application of computer programming to the reconstruction of a proto-language. In *International Conference on Computational Linguistics COLING 1969: Preprint No. 5*, Sånga Säby, Sweden.

Kuofeng Gao, Huanqia Cai, Qingyao Shuai, Dihong Gong, and Zhifeng Li. 2025. Embedding self-correction as an inherent ability in large language models for enhanced mathematical reasoning.

Satyam Goyal and Soham Dan. 2025. Iolbench: Benchmarking llms on linguistic reasoning.

Harald Hammarström, Robert Forkel, Martin Haspelmath, and Sebastian Bank. 2024. Glottolog 5.1. *Leipzig: Max Planck Institute for Evolutionary Anthropology.(Available online at glottolog.org, Accessed on 2025-02-06.)*, 10.

Jiayi He, Hehai Lin, Qingyun Wang, Yi Fung, and Heng Ji. 2025. Self-correction is more than refinement: A learning framework for visual and language reasoning tasks.

IOL. 2003-2025. International linguistics olympiad.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Zixuan Ke, Fangkai Jiao, Yifei Ming, Xuan-Phi Nguyen, Austin Xu, Do Xuan Long, Minzhi Li, Chengwei Qin, Peifeng Wang, Silvio Savarese, Caiming Xiong, and Shafiq Joty. 2025. A survey of frontiers in llm reasoning: Inference scaling, learning to reason, and agentic systems.

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

Wenzhe Li, Yong Lin, Mengzhou Xia, and Chi Jin. 2025. Rethinking mixture-of-agents: Is mixing different large language models beneficial?

Liang Lu, Peirong Xie, and David Mortensen. 2024. Semisupervised neural proto-language reconstruction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14715–14759, Bangkok, Thailand. Association for Computational Linguistics.

Atharva Naik, Kexun Zhang, Nathaniel Robinson, Aravind Mysore, Clayton Marr, Hong Sng Rebecca Byrnes, Anna Cai, Kalvin Chang, and David Mortensen. 2024. Can large language models code like a linguist?: A case study in low resource sound law induction.

Vlad A. Neacşu. 2024. *Linguistics Olympiad*. Number 13 in Textbooks in Language Sciences. Language Science Press, Berlin.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Bhargavi Paranjape, Scott Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Tulio Ribeiro. 2023. Art: Automatic multi-step reasoning and tool-use for large language models.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395.

Maja Popović. 2017. chrf++: words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612–618.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Gözde Gül Şahin, Yova Kementchedjhieva, Phillip Rust, and Iryna Gurevych. 2020. PuzzLing Machines: A Challenge on Learning From Small Data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1241–1254, Online. Association for Computational Linguistics.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools.

Yu-Fei Shih, Zheng-Lin Lin, and Shu-Kai Hsieh. 2025. Reasoning over the glyphs: Evaluation of llm's decipherment of rare scripts.

Eduardo Sánchez, Belen Alastruey, Christophe Ropers, Pontus Stenetorp, Mikel Artetxe, and Marta R. Costa-jussà. 2024. Linguini: A benchmark for language-agnostic linguistic reasoning.

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 *International Conference on Representation Learning*, volume 2024, pages 18955–18985.

Saujas Vaduguru, Aalok Sathe, Monojit Choudhury, and Dipti Sharma. 2021. Sample-efficient linguistic generalizations through program synthesis: Experiments with phonology problems. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 60–71, Online. Association for Computational Linguistics.

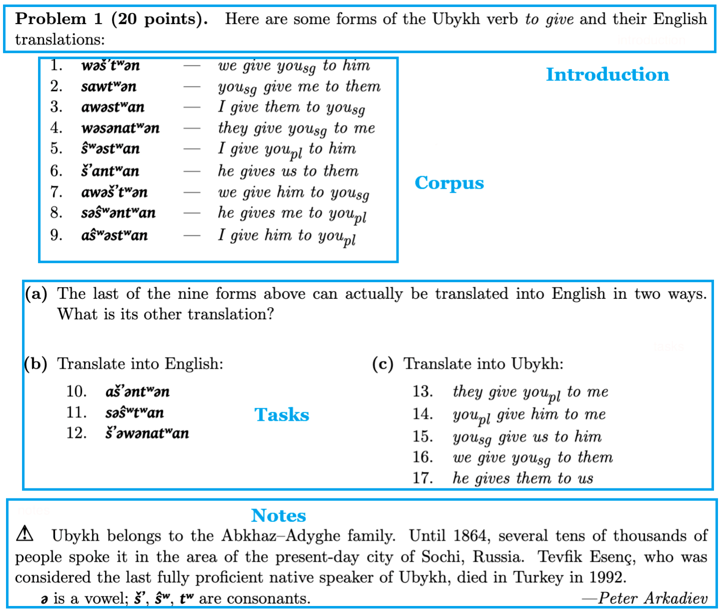
Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y Zou. 2025. Mixture-of-agents enhances large language model capabilities. In *International Conference on Representation Learning*, volume 2025, pages 33944–33963.

Mengsong Wu, Tong Zhu, Han Han, Xiang Zhang, Wenbiao Shao, and Wenliang Chen. 2025. Chain-of-tools: Utilizing massive unseen tools in the cot reasoning of frozen language models.

Kexun Zhang, Yee Choi, Zhenqiao Song, Taiqi He, William Yang Wang, and Lei Li. 2024. Hire a linguist!: Learning endangered languages in LLMs with in-context linguistic descriptions. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15654–15669, Bangkok, Thailand. Association for Computational Linguistics.

A IOL Problem Examples

A.1 Elements of an IOL Problem



Problem 1 (20 points). Here are some forms of the Ubykh verb *to give* and their English translations:

Introduction

1.	***wəš’tʷən***	—	*we give you$_{sg}$ to him*
2.	***sawtʷən***	—	*you$_{sg}$ give me to them*
3.	***awəstʷan***	—	*I give them to you$_{sg}$*
4.	***wəsənatʷən***	—	*they give you$_{sg}$ to me*
5.	***ŝʷəstʷan***	—	*I give you$_{pl}$ to him*
6.	***š’antʷan***	—	*he gives us to them*
7.	***awəš’tʷən***	—	*we give him to you$_{sg}$*
8.	***səŝʷəntʷan***	—	*he gives me to you$_{pl}$*
9.	***aŝʷəstʷan***	—	*I give him to you$_{pl}$*

Corpus

(a) The last of the nine forms above can actually be translated into English in two ways. What is its other translation?

(b) Translate into English:

10. ***aš’əntʷən***
11. ***səŝʷtʷan***
12. ***š’əwənatʷan***

Tasks

(c) Translate into Ubykh:

13. *they give you$_{pl}$ to me*
14. *you$_{pl}$ give him to me*
15. *you$_{sg}$ give us to him*
16. *we give you$_{sg}$ to them*
17. *he gives them to us*

Notes

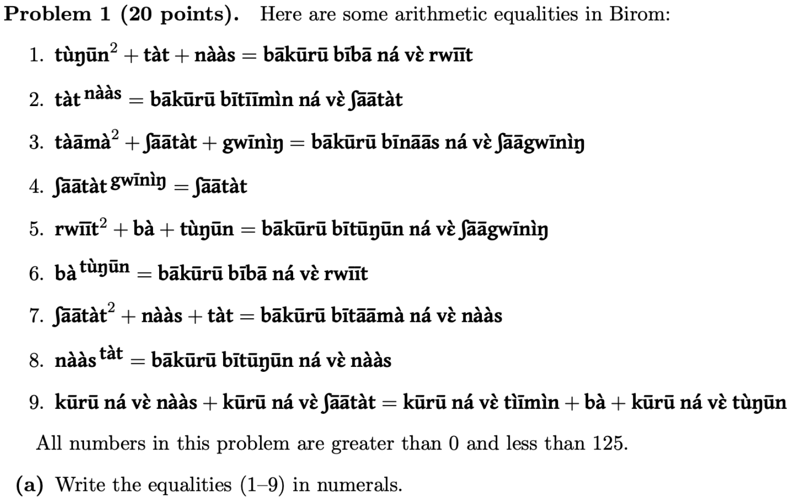
⚠ Ubykh belongs to the Abkhaz–Adyghe family. Until 1864, several tens of thousands of people spoke it in the area of the present-day city of Sochi, Russia. Tevfik Esenç, who was considered the last fully proficient native speaker of Ubykh, died in Turkey in 1992.

ə is a vowel; ***š’***, ***ŝʷ***, ***tʷ*** are consonants. —*Peter Arkadiev*

Figure 3: An IOL Problem with the four parts: **Introduction** provides information about the language(s) featured in the problem; **Corpus** contains the examples based on which the tasks should be solved. **Tasks** follows the corpus and typically includes translation between the target language and English, correspondences of randomly arranged items, among other types of tasks; **Notes** provide data about the language featured in the problem, relevant phonetic/orthographic information, and details about specific words. Any additional information crucial to solving the problem will be included in the **introduction** and **notes** sections.

+

A.2 More Examples on Diversity in Problem



Problem 1 (20 points). Here are some arithmetic equalities in Birom:

1. **tùŋūn**^{2} + **tàt** + **nààs** = **bākūrū bībā ná vὲ rwīīt**
2. **tàt**$^{\textbf{nààs}}$ = **bākūrū bītīīmìn ná vὲ ʃāātàt**
3. **tàāmà**^{2} + **ʃāātàt** + **gwīnìŋ** = **bākūrū bīnāās ná vὲ ʃāāgwīnìŋ**
4. **ʃāātàt**$^{\textbf{gwīnìŋ}}$ = **ʃāātàt**
5. **rwīīt**^{2} + **bà** + **tùŋūn** = **bākūrū bītūŋūn ná vὲ ʃāāgwīnìŋ**
6. **bà**$^{\textbf{tùŋūn}}$ = **bākūrū bībā ná vὲ rwīīt**
7. **ʃāātàt**^{2} + **nààs** + **tàt** = **bākūrū bītāāmà ná vὲ nààs**
8. **nààs**$^{\textbf{tàt}}$ = **bākūrū bītūŋūn ná vὲ nààs**
9. **kūrū ná vὲ nààs** + **kūrū ná vὲ ʃāātàt** = **kūrū ná vὲ tììmìn** + **bà** + **kūrū ná vὲ tùŋūn**

All numbers in this problem are greater than 0 and less than 125.

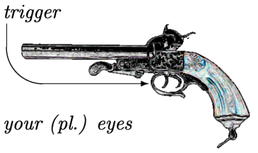
(a) Write the equalities (1–9) in numerals.

Figure 4: Problem 1 (IOL 2017)

Problem 2 (20 points). Here are some words and word combinations in Abui and their English translations in arbitrary order:

1. **abang**
2. **atáng heya**
3. **bataa hawata**
4. **dekafi**
5. **ebataa hatáng**
6. **ekuda hawata**
7. **falepak hawei**
8. **hatáng hamin**
9. **helui**
10. **maama hefalepak**
11. **napong**
12. **rièng**
13. **ritama**
14. **riya hatáng**
15. **tama habang**
16. **tamin**
17. **tefe hawei**

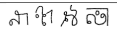
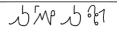
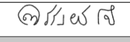
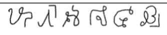
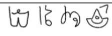
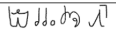
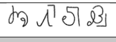
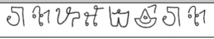
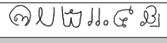
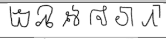
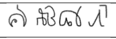
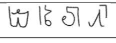
a. *his fingertip*
b. *your (sg.) branch*
c. *my face*
d. *one's own rope*
e. *your (sg.) shoulder*
f. *your (pl.) mother's hand*
g. *our pigs' ears*
(the ear of the pig of each of us)
h. *father's pistol*
i. *your (sg.) horse's neck*
j. *trigger*

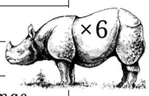


k. *your (pl.) eyes*
l. *our noses*
(the nose of each of us)
m. *his knife*
n. *seashore*
o. *upper part of a tree*
p. *your (sg.) thumb*
q. *your (pl.) sea*

Figure 5: Problem 2 (IOL 2017)

Problem 4 (20 points). Here are some word combinations in Laven written in the Khom script and in phonetic transcription and their English translations:

1		**praj trie**	*to wake up the wife*
2		**ca:k caj**	*from the heart/mind/soul*
3	?	**taw bɛ:**	*to see the raft*
4		**krɨət blaw**	*to scratch the thigh*
5		**plaj prɨət**	*banana*
6	?	?	*three bananas*
7		?	*six rhinoceros*
8		?	*four hands of bananas*
9		?	*?*
10	?	**cie pʌh la:**	*seven sheets of paper*
11		?	*aubergine/eggplant leaf*
12		?	*two aubergines/eggplants*
13		**plaj hnat pʌh plaj**	*seven pineapples*
14		**kruat pɛ: to:**	*three bees*
15		**la: prɨət traw la:**	*?*
16	?	**kə:r bə:r to:**	*two doves*
17		**bla:k puan ka:**	*four carp*
18		**piet traw pla:**	*six knives*
19		**bə:r ka:**	*?*
20		?	*four blades*



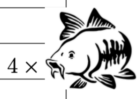
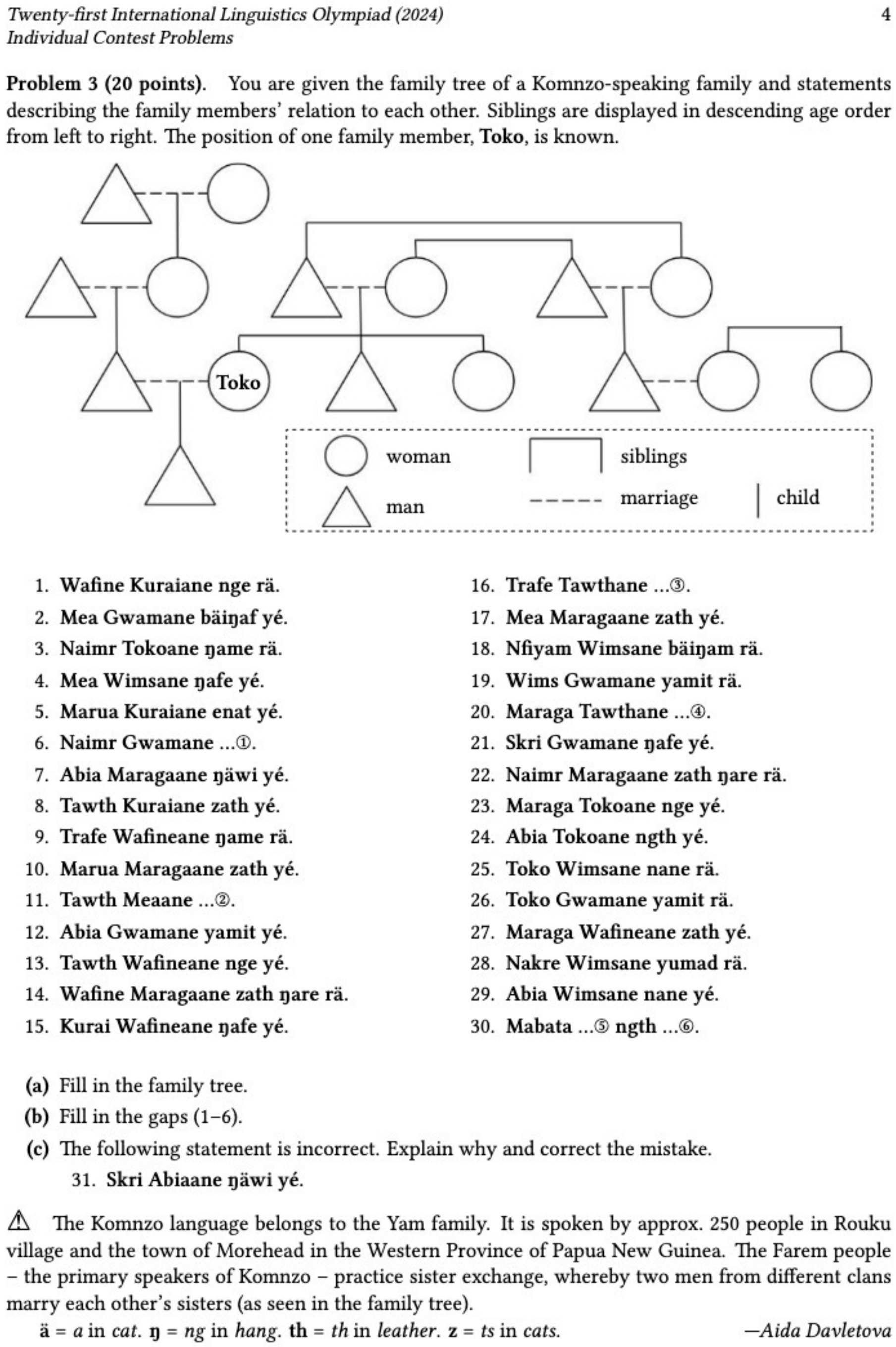


Figure 6: Problem 4 (IOL 2017)

A.3 Example on Kinship Problem&Graph Transcription



Twenty-first International Linguistics Olympiad (2024) 4
Individual Contest Problems

Problem 3 (20 points). You are given the family tree of a Komnzo-speaking family and statements describing the family members' relation to each other. Siblings are displayed in descending age order from left to right. The position of one family member, **Toko**, is known.

1. **Wafine Kuraiane nge rä.**
2. **Mea Gwamane bäiŋaf yé.**
3. **Naimr Tokoane ŋame rä.**
4. **Mea Wimsane ŋafe yé.**
5. **Marua Kuraiane enat yé.**
6. **Naimr Gwamane** ...①.
7. **Abia Maragaane ŋäwi yé.**
8. **Tawth Kuraiane zath yé.**
9. **Trafe Wafineane ŋame rä.**
10. **Marua Maragaane zath yé.**
11. **Tawth Meaane** ...②.
12. **Abia Gwamane yamit yé.**
13. **Tawth Wafineane nge yé.**
14. **Wafine Maragaane zath ŋare rä.**
15. **Kurai Wafineane ŋafe yé.**
16. **Trafe Tawthane** ...③.
17. **Mea Maragaane zath yé.**
18. **Nfiyam Wimsane bäiŋam rä.**
19. **Wims Gwamane yamit rä.**
20. **Maraga Tawthane** ...④.
21. **Skri Gwamane ŋafe yé.**
22. **Naimr Maragaane zath ŋare rä.**
23. **Maraga Tokoane nge yé.**
24. **Abia Tokoane ngth yé.**
25. **Toko Wimsane nane rä.**
26. **Toko Gwamane yamit rä.**
27. **Maraga Wafineane zath yé.**
28. **Nakre Wimsane yumad rä.**
29. **Abia Wimsane nane yé.**
30. **Mabata** ...⑤ **ngth** ...⑥.

(a) Fill in the family tree.

(b) Fill in the gaps (1–6).

(c) The following statement is incorrect. Explain why and correct the mistake.

31. **Skri Abiaane ŋäwi yé.**

⚠ The Komnzo language belongs to the Yam family. It is spoken by approx. 250 people in Rouku village and the town of Morehead in the Western Province of Papua New Guinea. The Farem people – the primary speakers of Komnzo – practice sister exchange, whereby two men from different clans marry each other's sisters (as seen in the family tree).

ä = *a* in *cat*. **ŋ** = *ng* in *hang*. **th** = *th* in *leather*. **z** = *ts* in *cats*. *—Aida Davletova*

Figure 7: Original Problem 3 in 2024.

Transcription of the Family Tree

- Man 1 and Woman 1 are married. Their child is Woman 2.
- Man 2 and Woman 2 are married. Their child is Man 3.
- Man 3 and Woman 3 are married. Their child is Man 4.
- Woman 3 is Toko.
- Man 5 and Woman 4 are married. Their children are Woman 3, Man 6 and Woman 5, from oldest to youngest.
- Man 5 and Woman 6 are siblings. The former is older.
- Woman 4 and Man 7 are siblings. The former is older.
- Man 7 and Woman 6 are married. Their child is Man 8.
- Man 8 and Woman 7 are married.
- Woman 7 and Woman 8 are siblings. The former is older.

B Prompt Template for Reasoning Process Generation

The following Python template was used to generate reasoning chains for IOL problems:

```
## Prompt:
As an expert in linguistics solve the following problem. Given the following IOL problem and its answer, generate a detailed, step-by-step chain of thoughts that could specifically and reasonably lead to the answer. Focus on the reasoning process, essential linguistic rules, logical deductions, and the final solution. Make your whole output into a markdown file.

## Problem:
{problem_text}

## Solution:
{solution_text}

## Your response:
```

C The Classification Framework for Problems

Category	Tag
Subject	Compounding, Morphology, Numbers, Phonology and Phonetics, Semantics, Syntax, Writing System
Type	Rosetta, Match-up, Monolingual, Pattern, Computational, Text
Theme	Classical, Comparative, Encrypted, Kinship, Maps, Mystery, MFL,[1] Senses and Feelings, Stories, Poetry, No Theme

[1] MFL: These questions involve languages commonly taught in secondary school MFL departments, or those closely related (e.g., Romance and Germanic languages).

Table 1: Typological Annotation Category

Sub-problems	Subject	Type	Language	Speakers	glottocode	Language Family
2	Numbers	Pattern	Egyptian Arabic	68,000,000	egyp1253	Semitic

Table 2: Example of Typological Annotation: Problem 2 in 2003

C.1 Classification Criteria

The following categories and the classification criteria are modified from those of UKLO[9].

- **Subjects** –For a given subject to appear in the classification, at least two rules in the solution must be of that type.
  - **Compounding**: The problems mainly focus on deducing the dictionary meanings of words by analyzing how the meaning changes when different word components are combined.
  - **Morphology**: The problems primarily require understanding how morphemes (the smallest units of meaning) combine to form grammatical words.
  - **Numbers**: The problems are centered on understanding the structure and formation of numerals and numeral expressions.
  - **Phonology and Phonetics**: The problems focus on the sounds of a language and how they are organized. Phonology deals with sound systems within specific languages and in general, while phonetics studies the nature, production, and perception of speech sounds, independent of any particular language.

[9] https://www.uklo.org/technical-information/

  - **Semantics**: The problems emphasize understanding how meaning influences language, especially how meaning shapes grammar and how different languages express the same concepts with different words.
  - **Syntax**: The problems focus on understanding how words combine to form phrases and sentences.
  - **Writing System**: The problems involve analyzing writing systems, including both the use of the Latin alphabet in various languages and other scripts.

- **Problem Type**

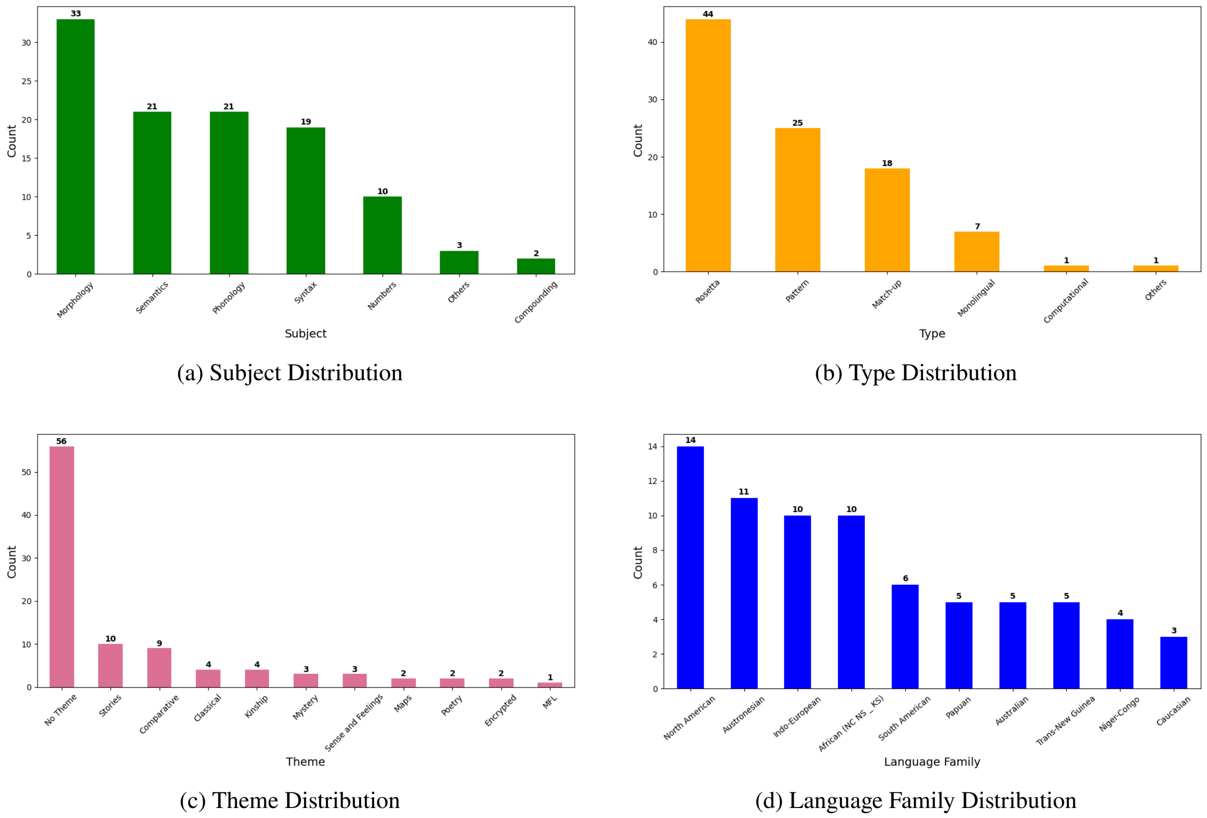
  - **Rosetta**: The problems consist of sets of corresponding words or phrases across different languages or writing systems, with most pairings provided. Some elements may be missing, creating gaps that need to be filled. Solving the task requires generating new correspondences, typically translations.
  - **Match-up**: The problems consist of sets of corresponding words or phrases across multiple languages or writing systems, with only a few pairings given. Some words may not belong to any set, but it still qualifies as a match-up. The task involves identifying new correspondences, usually translations.
  - **Monolingual**: The problems are texts in an unfamiliar language (or equivalent writing system), generally without direct translations or transliterations, except perhaps for one or two words. To solve the task, you must translate the text from the unknown language.
  - **Pattern**: The problems consist of words or groups of word forms or cognates that follow a certain pattern, though there may be exceptions. To solve the task, you must either generate other examples that fit the pattern or identify exceptions, without relying on translation as in Rosetta tasks.
  - **Computational**: The problems include a description of a computational or logical system. Solving the problem involves analyzing and implementing the system according to the given rules.
  - **Text**: The problems consist of full texts in different languages or scripts, without being broken down into smaller parts. To solve the task, you must infer linguistic rules using context and other cues.

- **Theme**

  - **Classical**: These problems feature languages that were primarily spoken around a thousand years ago or earlier.
  - **Comparative**:These problems involve comparing either related languages or different historical stages of a single language.
  - **Encrypted**: These problems involve deciphering an encoded message in English.
  - **Kinship**: These problems focus on understanding how different languages and cultures describe family relationships and naming systems.
  - **Maps**: These problems explore how various languages express and conceptualize directions and spatial orientation.
  - **Mystery**: These problems include a mystery element that draws on general or world knowledge, often involving content beyond linguistics.
  - **MFL**: These problems involve languages commonly taught in secondary school modern foreign language (MFL) departments, or closely related languages (e.g., those from the Romance or Germanic families).
  - **Senses and Feelings**: These problems examine linguistic expressions related to emotions or sensory experiences (e.g., smells, sounds).

- **Stories**: These problems either contain a narrative storyline or feature one or more fictional characters. They use storytelling to create engaging contexts for linguistic analysis, often drawing from literary traditions.
- **Poetry**: These problems revolve around the structure and features of poetic language.
- **No Theme (N/A)**: These problems focus on core linguistic topics without any specific thematic context.

D Preliminary Analysis of IOL Problems.



(a) Subject Distribution

(b) Type Distribution

(c) Theme Distribution

(d) Language Family Distribution

Figure 8: Statistical distributions of various features in the IOL problems dataset.

E Heatmap: Subject vs Type Distribution

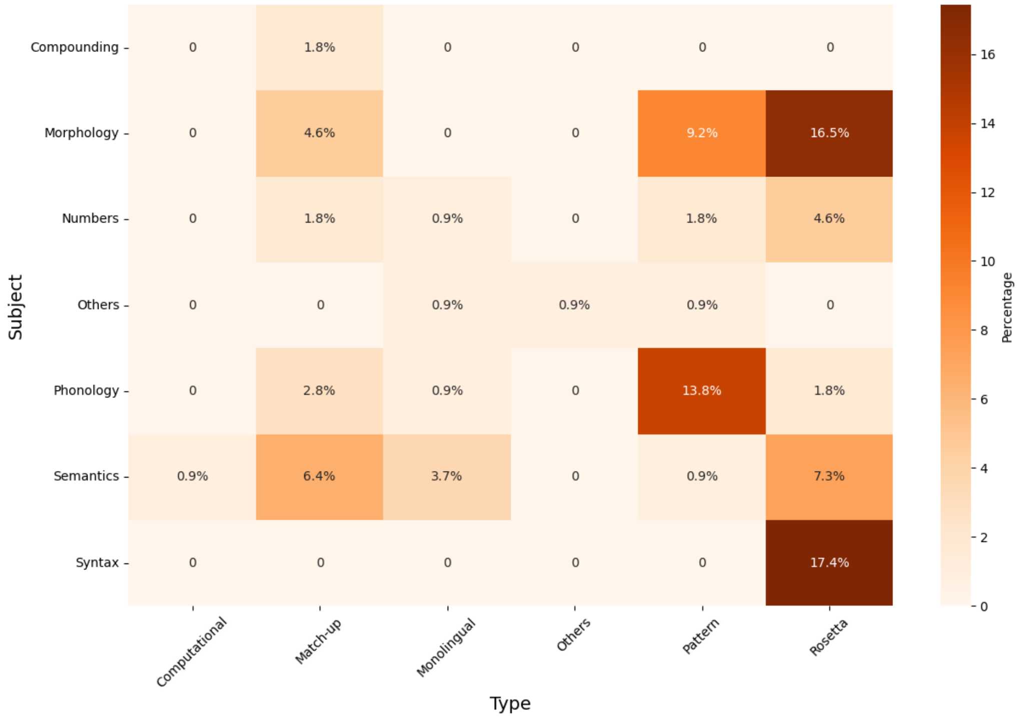


Figure 9: Subject vs Type Distribution

F Heatmap: Subject vs Language Family Distribution

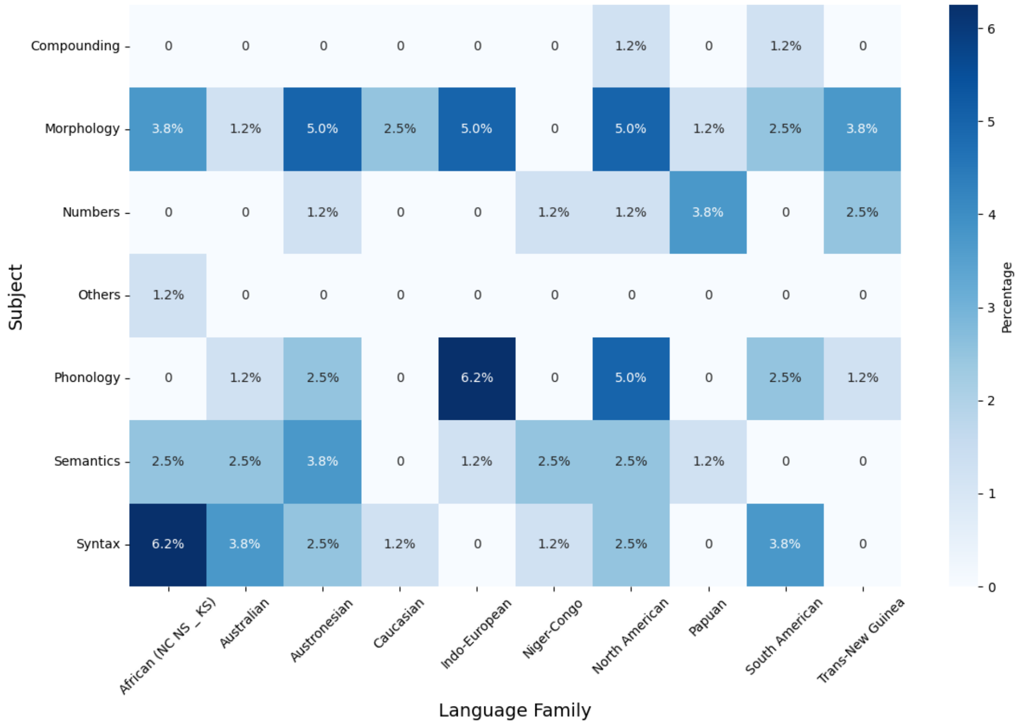


Figure 10: Subject vs Top 10 Language Family Distribution

G Heatmap: Type vs Language Family Distribution

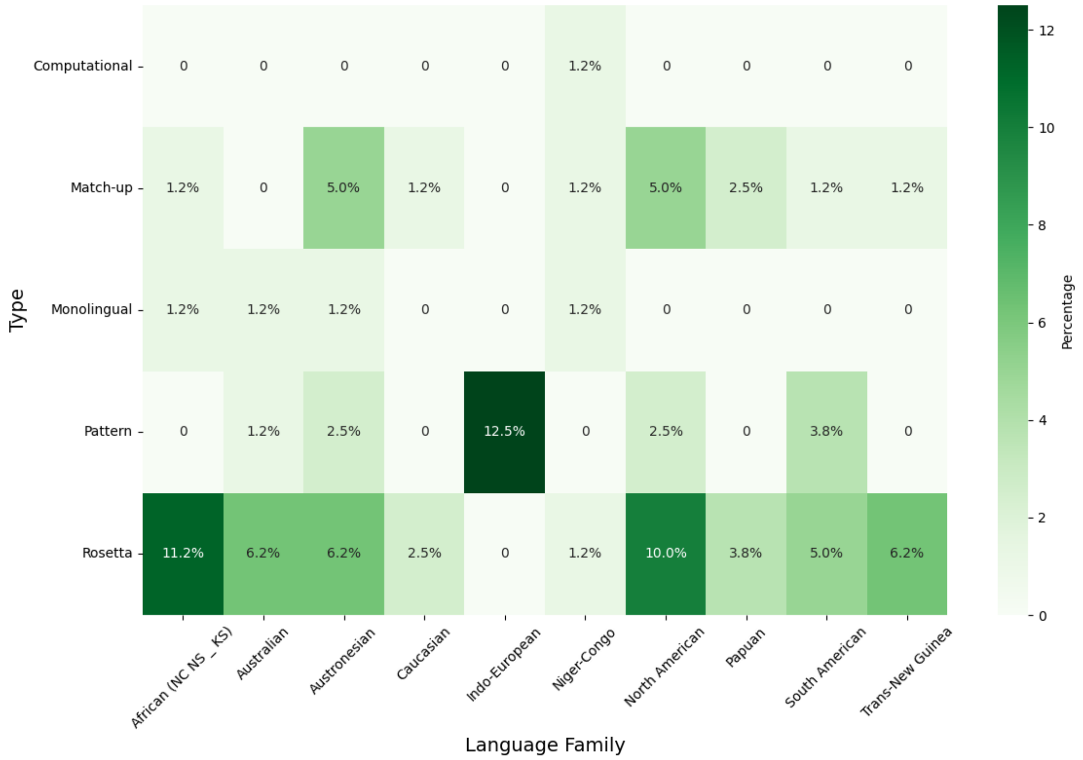


Figure 11: Type vs Top 10 Language Family Distribution

H System Prompt for Model Reasoning Evaluation

```
system_prompt = """Given the evaluation rules and metrics for model reasoning of IOL problems, consider the golden reasoning reference, and evaluate the target model reasoning with the metrics of five dimensions.
evaluation rules and metrics (5-score):
{metrics}

scoring_:
{scoring}

golden reasoning reference:
{golden_reasoning_reference}

target model reasoning:
{model_reasoning}
"""

metrics = """
### Metrics and Descriptions (Bullet Points)
 (i) 3.1 Information Extraction & Structuring
  * **Stepwise Logical Validity Score (SLVS)**: Measures whether each reasoning step is logically valid and aligned with the golden reasoning reference (GRR).
  * **Information Structuring Completeness (ISC)**: Measures completeness of extracted key information and its structure compared to GRR.
  ... [TRUNCATED FOR BREVITY IN PAPER]
"""

scoring = """
## Reasoning Quality Evaluation -- Scoring Rubric (5 Points per Metric)

| **Dimension** | **Metric** | **Score 5 (Excellent)** | **Score 3 (Acceptable)** | **Score 1 (Poor)** |
| ----- | ----- | ----- | ----- | ----- |
| **(i) 3.1 Information Extraction & Structuring** | **SLVS**   | All reasoning steps are logically valid and follow GRR structure | Minor logical flaws or omissions; generally coherent | Major logical errors, incoherent or illogical steps | |  | **ISC**    | Extracts and structures all key information as per GRR | Extracts partial or incomplete key information | Fails to extract/structure key information |
... [TRUNCATED FOR BREVITY IN PAPER]
"""

golden_reasoning_reference = """
# Your response:

# Chain of Thought: Solving the Swift News Linguistics Problem

... [TRUNCATED FOR BREVITY IN PAPER]
"""

target_model_reasoning = """
**Solving the Linguistic Puzzle**

... [TRUNCATED FOR BREVITY IN PAPER]
"""
```

I FLORES-200 Multilingual Evaluation Details

Dataset preparation and experimental design. We combine the dev and devtest splits for a total of 2009 sentences that are available in **204** languages. We then use the ISO 639-3 language code and the ISO 15924 script code to identify the Glottocode and the script used for each language, respectively. For example, the column name for Bashkir translations written in Cyrillic is `sentence_bak_Cyrl`. To align the dataset with the Glottolog taxonomy, we mapped all language identifiers to their corresponding Glottolog codes. We noted that five ISO 639-3 codes from the dataset (i.e., `srd`, `est`, `kon`, `zho`, `grn`) were not directly linked to a Glottolog entry. We identified suitable entries manually. How we mapped these languages can be found in Table 6. In total, we have 204 languages and script combinations.[10] Next, we take the first **10** English sentences and their translations for a total of **2030** English-to-Target Language pairs.

We evaluate Gemini-2.5-flash with `temperature=0.1` and `thinking budget=0` by translating from two directions: *English-to-Target* ($E \rightarrow T$) and *Target-to-English* ($T \rightarrow E$). We use the following $E \rightarrow T$ prompt when eliciting a response from the model:

```
Translate the following sentence from English to {target_lang} using
the {script} script:
Input: {input_sentence}
```

We use the following $T \rightarrow E$ prompt:

```
Translate the following sentence {target_lang} to English:
Input: {input_sentence}
```

Language	**Glottocode**	**Class**	**Missing** ($E \rightarrow T$)	**Missing** ($T \rightarrow E$)	**Total Missing**
Tamasheq	tama1365	0	7	1	8
Nuer	nuer1246	0	6	2	8
Kabiyé	kabi1261	0	7	0	7
Southwestern Dinka	sout2832	—	6	1	7
Central Kanuri	cent2050	0	4	2	6
Fon	fonn1241	0	5	0	5
Chokwe	chok1245	—	2	1	3
Umbundu	umbu1257	0	3	0	3
Kamba (Kenya)	kamb1297	0	2	0	2
Sango	sang1328	1	2	0	2
South-Central Koongo	koon1244	1	2	0	2
Kimbundu	kimb1241	0	2	0	2
Bambara	bamb1269	1	2	0	2
Dyula	dyul1238	0	2	0	2
Mossi	moss1236	0	4	0	4
Southern Jinghpaw	kach1280	0	4	0	4
Shan	shan1277	0	4	0	4
Acehnese	achi1257	1	1	0	1
Ewe	ewee1241	1	1	0	1
Dzongkha	dzon1239	1	1	0	1
Central Aymara	cent2142	—	1	0	1
Ayacucho Quechua	ayac1239	—	1	0	1
Luba-Lulua	luba1249	0	1	0	1
Kabyle	kaby1243	1	1	0	1
Guarani	east2555	1	1	0	1
Wolof	nucl1347	2	1	0	1
Grand Total			**73**	**7**	**80**

Table 3: Counts of missing LLM Outputs by language and direction. *Class* refers to the taxonomy introduced in Joshi et al. (2020) in which 0 indicates extremely limited resources and 5 indicates an abundance of resources. "–" means that the language was not found in the taxonomy.

[10] 196 unique languages while Acehnese, Minangkabau, Banjar, Central Kanuri, Tamasheq, Standard Arabic, Kashmiri, and Mandarin each have two scripts.

With the LLM translating in two directions, we obtain 3800 responses; however, 80 responses are empty with the majority of them originating from the $E \rightarrow T$ task. We will first examine these failures.

The LLM often fails to output any text for low resource languages. From the results in Table 3 we can see that the data strongly suggests that the model's failure to generate output is directly linked to data resource scarcity. The *Class* column refers to the taxonomy introduced in Joshi et al. (2020) where Class 0 languages have a dearth of resources while the Class 5 languages are at the opposite end of the spectrum.[11] The vast majority of missing outputs are concentrated in languages designated as Class 0 (e.g., Tamasheq, Nuer, Kabiyé), which represents the lowest-resource tier in our dataset. "–" means that the language was not found in the taxonomy.

Furthermore, the model fails far more frequently in the *English-to-Target* direction (73 instances) than in the Target-to-English direction (7 instances). This indicates that the primary challenge is not the model's ability to process or analyze the target languages (i.e., $T \rightarrow E$), but rather its capacity to reliably *generate* text in them (i.e., $E \rightarrow T$). This strongly suggests limited training data in the target language. This conclusion is reinforced by the performance on higher-resourced languages. We will now examine the overall translation quality of the outputs.

LLM performance is heavily influenced by translation direction, language family, and resource availability. We use chrF (Popović, 2015) instead of chrF+ or chrF++ (Popović, 2017) because the former is language independent and tokenization independent, which is needed when many languages found in FLORES-200 may not have a robust tokenizer or even have one readily available. chrF measures translation quality by calculating character-level n-gram overlap F-score between the machine translation and the human translation. The latter two introduces word unigram and bigram overlap into the equation. We use the implementation provided by Hugging Face with default parameters,[12] which adopts the implementation from sacreBLEU (Post, 2018)[13] but with a slightly different input format.

Direction	Mean chrF Score	Correlation with Class (ρ)
$E \rightarrow T$	43.92	0.598
$T \rightarrow E$	64.27	0.466

Table 4: Mean chrF scores and their Spearman's correlation (ρ) with resource class for each translation direction.

Worth noting is the direction where the model is worse on average ($E \rightarrow T$) is also the direction where performance is more strongly influenced by resource availability (higher correlation, $\rho = 0.598$). This suggests that while translating into English has a relatively high performance floor, the model's ability to generate text in other languages is both lower on average and more vulnerable to data scarcity. Figure 12 paints a similar picture in which lower resource classes predictably have worse performance compared to languages with more resources. We also see that translating from English to another language exacerbates the problem.

To also see how language family and script influence translation quality we used three separate one-way ANOVAs for each translation direction ($E \rightarrow T$ and $T \rightarrow E$). The results, summarized in Table 5, indicate that both **family** and **class** have a large and highly significant effect on performance in both directions (all $p < .001$). In contrast, **script** was not found to be a statistically significant predictor of chrF score in either analysis.

The analysis reveals an important asymmetry in the influence of resource class. While significant in both cases, **class** accounts for a larger portion of the variance in $E \rightarrow T$ scores ($\eta_p^2 = .412$) than in $T \rightarrow E$ scores ($\eta_p^2 = .381$).

[11] Because the language name to class list from Joshi et al. does not use an ISO 639-3 or Glottocode, we can only use the name to identify which language is paired with which Glottocode. We only assign classes for unambiguous language names. For example, while "khmer" is found in the language to class list, we do not join it with "Central Khmer." There are 30 languages without an assigned Resource Class.

[12] https://huggingface.co/spaces/evaluate-metric/chrf

[13] https://github.com/mjpost/sacreBLEU#chrf--chrf

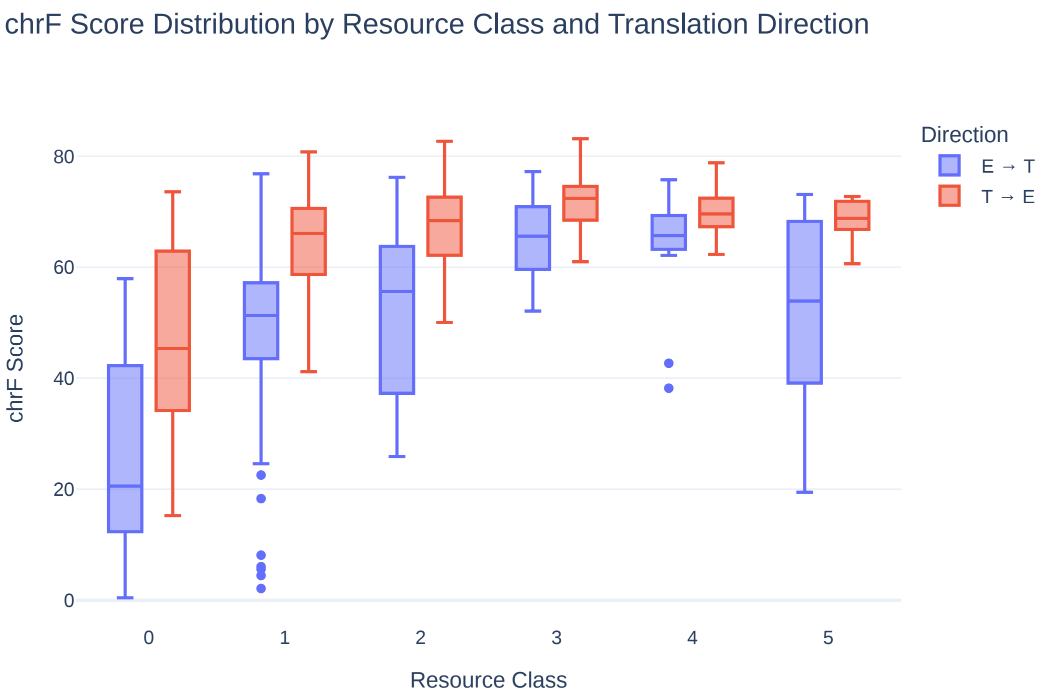


Figure 12: Comparison of chrF score distributions for English-to-Target ($E \to T$) and Target-to-English ($T \to E$) translations, grouped by resource class. The plot shows a clear positive trend where quality increases with resource availability, with the $T \to E$ direction consistently outperforming the $E \to T$ direction. Boxes represent the interquartile range, and points show individual languages that fall beyond the lower fence.

	$E \to T$		$T \to E$	
Factor	**Effect Size (η_p^2)**	**p-value**	**Effect Size (η_p^2)**	**p-value**
Family	0.409	$< .001$	0.515	$< .001$
Class	0.412	$< .001$	0.381	$< .001$
Script	0.174	.265	0.125	.740

Table 5: Summary of One-Way ANOVA results showing the influence of each factor on chrF scores. Effect sizes are given as partial eta-squared (η_p^2).

This illustrates that processing low-resource languages still proves to be a challenge for even the most powerful of models. FLORES-200 only covers a small fraction of the world's languages and were chosen carefully based on several considerations, such as having a presence on Wikipedia. This limitation with processing low-resource languages will only be more pronounced when we examine other languages with even fewer resources. The results for each language can be found in Table 7 as well as additional figures for script and language family-level scores in Section L of the Appendix.

Given that these results stem from a single experimental iteration, they should be interpreted as preliminary. Nevertheless, they provide strong evidence of the lopsided distribution of data resources among the world's languages and imbalanced performance across languages for today's SOTA LLMs, which warrants further investigation.

J Resolution of Ambiguous ISO 639-3 to Glottocode Mappings

Table 6: Resolution of ambiguous source ISO 639-3 codes to specific language varieties and their corresponding Glottocode.

Language Mapping Details	
`srd`	**Language:** Sardinian **ISO → Glottocode:** None → `sard1257` **Justification:** Top-level family node.
`est`	**Language:** Estonian **ISO → Glottocode:** `ekk` → `esto1258` **Justification:** Primary language entry.
`kon`	**Language:** South-Central Kongo **ISO → Glottocode:** `kng` → `koon1244` **Justification:** Known as Kongo in World Atlas of Language Structures (WALS).
`zho`	**Language:** Mandarin **ISO → Glottocode:** `cmn` → `mand1415` **Justification:** Most populous variety.
`grn`	**Language:** Eastern Bolivian Guaraní **ISO → Glottocode:** `gui` → `east2555` **Justification:** Guaraní categorized as Class 1 in Joshi et al. (2020), which aligns more with Ethnologue's Digital Language Support classification of "Ascending" for the language.

K Language-Level chrF Translation Scores for Gemini-2.5-Flash on FLORES-200

Table 7: Performance results by language, including chrF scores, sample counts, and resource class.

Language (glottocode_Script)	$E \rightarrow T$ chrF	$T \rightarrow E$ chrF	Family	Class	Samples ($E \rightarrow T$ / $T \rightarrow E$)
Acehnese (achi1257_Arabic)	6.05	49.46	Austronesian	1	9 / 10
Acehnese (achi1257_Latin)	46.42	71.11	Austronesian	1	10 / 10
Afrikaans (afri1274_Latin)	73.91	83.15	Indo-European	3	10 / 10
Akan (akan1250_Latin)	37.78	49.00	Atlantic-Congo	1	10 / 10
Amharic (amha1245_Ethiopic (Ge'ez))	35.85	70.71	Afro-Asiatic	2	10 / 10
Assamese (assa1263_Bengali)	48.08	67.72	Indo-European	1	10 / 10
Asturian-Leonese-Cantabrian (astu1245_Latin)	69.93	73.94	Indo-European	1	10 / 10
Awadhi (awad1243_Devanagari (Nagari))	41.45	67.04	Indo-European	0	10 / 10
Ayacucho Quechua (ayac1239_Latin)	37.25	53.34	Quechuan	–	9 / 10
Balinese (bali1278_Latin)	44.79	61.53	Austronesian	0	10 / 10
Bambara (bamb1269_Latin)	2.12	41.72	Mande	1	8 / 10
Banjar (banj1239_Arabic)	4.46	53.69	Austronesian	1	10 / 10
Banjar (banj1239_Latin)	51.99	60.64	Austronesian	1	10 / 10
Bashkir (bash1264_Cyrillic)	56.01	68.67	Turkic	1	10 / 10
Basque (basq1248_Latin)	64.81	67.00	Unknown	4	10 / 10
Belarusian (bela1254_Cyrillic)	52.41	60.98	Indo-European	3	10 / 10
Bemba (Zambia) (bemb1257_Latin)	43.05	60.86	Atlantic-Congo	0	10 / 10
Bengali (beng1280_Bengali)	59.45	68.50	Indo-European	3	10 / 10
Bhojpuri (bhoj1244_Devanagari (Nagari))	44.14	62.46	Indo-European	1	10 / 10
Bosnian Standard (bosn1245_Latin)	67.72	70.89	Indo-European	3	10 / 10
Buginese (bugi1244_Latin)	35.98	48.74	Austronesian	1	10 / 10
Bulgarian (bulg1262_Cyrillic)	76.45	76.70	Indo-European	3	10 / 10
Burmese (nucl1310_Myanmar (Burmese))	53.93	68.35	Sino-Tibetan	1	10 / 10
Catalan (stan1289_Latin)	67.90	72.33	Indo-European	4	10 / 10
Cebuano (cebu1242_Latin)	65.84	80.13	Austronesian	3	10 / 10
Central Aymara (cent2142_Latin)	31.09	44.91	Aymaran	–	9 / 10
Central Kanuri (cent2050_Arabic)	2.31	15.26	Saharan	0	10 / 8
Central Kanuri (cent2050_Latin)	8.76	32.46	Saharan	0	6 / 10
Central Khmer (cent1989_Khmer)	43.45	73.44	Austroasiatic	–	10 / 10
Central Kurdish (cent1972_Arabic)	51.29	67.85	Indo-European	–	10 / 10
Central Moroccan Berber (cent2194_Tifinagh (Berber))	26.34	45.68	Afro-Asiatic	0	10 / 10
Chhattisgarhi (chha1249_Devanagari (Nagari))	50.58	70.19	Indo-European	–	10 / 10
Chokwe (chok1245_Latin)	19.24	28.74	Atlantic-Congo	–	8 / 9
Crimean Tatar (crim1257_Latin)	45.90	70.46	Turkic	1	10 / 10
Croatian Standard (croa1245_Latin)	62.14	69.88	Indo-European	4	10 / 10
Czech (czec1258_Latin)	63.42	73.96	Indo-European	4	10 / 10
Danish (dani1285_Latin)	77.23	75.40	Indo-European	3	10 / 10
Dari (dari1249_Arabic)	42.70	65.11	Indo-European	4	10 / 10
Dutch (dutc1256_Latin)	66.63	68.29	Indo-European	4	10 / 10
Dyula (dyul1238_Latin)	16.31	33.30	Mande	0	8 / 10
Dzongkha (dzon1239_Tibetan)	33.37	50.97	Sino-Tibetan	1	9 / 10
East Latvian (east2282_Latin)	45.29	73.02	Indo-European	–	10 / 10
Eastern Armenian (nucl1235_Armenian)	61.44	71.83	Indo-European	1	10 / 10
Eastern Panjabi (panj1256_Gurmukhi)	56.52	73.75	Indo-European	–	10 / 10
Eastern Yiddish (east2295_Hebrew)	42.70	83.12	Indo-European	–	10 / 10
Egyptian Arabic (egyp1253_Arabic)	52.11	65.85	Afro-Asiatic	3	10 / 10
Esperanto (espe1235_Latin)	66.90	76.29	Artificial Language	1	10 / 10
Estonian (esto1258_Latin)	61.18	67.24	Uralic	3	10 / 10
Ewe (ewee1241_Latin)	36.73	49.73	Atlantic-Congo	1	9 / 10
Faroese (faro1244_Latin)	64.14	77.47	Indo-European	1	10 / 10
Fijian (fiji1243_Latin)	50.32	55.60	Austronesian	1	10 / 10
Finnish (finn1318_Latin)	66.57	68.23	Uralic	4	10 / 10
Fon (fonn1241_Latin)	7.64	23.20	Atlantic-Congo	0	5 / 10
French (stan1290_Latin)	73.10	70.06	Indo-European	5	10 / 10
Friulian (friu1240_Latin)	61.78	67.92	Indo-European	1	10 / 10
Galician (gali1258_Latin)	65.03	70.14	Indo-European	3	10 / 10
Ganda (gand1255_Latin)	42.86	56.15	Atlantic-Congo	1	10 / 10
Georgian (nucl1302_Georgian (Mkhedruli))	56.55	63.11	Kartvelian	3	10 / 10
German (stan1295_Latin)	71.48	72.08	Indo-European	5	10 / 10

Continued on next page

Table 7 – continued from previous page

Language **(glottocode_Script)**	$E \rightarrow T$ **chrF**	$T \rightarrow E$ **chrF**	**Family**	**Class**	**Samples** ($E \rightarrow T$ / $T \rightarrow E$)
Gilit Mesopotamian Arabic (meso1252_Arabic)	51.31	66.27	Afro-Asiatic	–	10 / 10
Guarani (east2555_Latin)	30.45	60.51	Tupian	1	9 / 10
Gujarati (guja1252_Gujarati)	49.30	70.17	Indo-European	1	10 / 10
Haitian (hait1244_Latin)	62.81	69.51	Indo-European	2	10 / 10
Halh Mongolian (halh1238_Cyrillic)	54.87	71.44	Mongolic-Khitan	0	10 / 10
Hausa (haus1257_Latin)	61.93	67.27	Afro-Asiatic	2	10 / 10
Hausa States Fulfulde (nige1253_Latin)	23.36	34.24	Atlantic-Congo	–	10 / 10
Hindi (hind1269_Devanagari (Nagari))	64.11	69.33	Indo-European	4	10 / 10
Hungarian (hung1274_Latin)	69.67	71.54	Uralic	4	10 / 10
Icelandic (icel1247_Latin)	65.36	69.58	Indo-European	2	10 / 10
Igbo (nucl1417_Latin)	50.62	64.71	Atlantic-Congo	1	10 / 10
Iloko (ilok1237_Latin)	56.05	69.03	Austronesian	1	10 / 10
Irish (iris1253_Latin)	64.73	77.31	Indo-European	2	10 / 10
Italian (ital1282_Latin)	62.85	64.59	Indo-European	4	10 / 10
Japanese (nucl1643_Japanese)	53.92	72.73	Japonic	5	10 / 10
Javanese (java1254_Latin)	64.70	71.11	Austronesian	1	10 / 10
Kabiyé (kabi1261_Latin)	0.44	39.03	Atlantic-Congo	0	3 / 10
Kabuverdianu (kabu1256_Latin)	58.01	75.68	Indo-European	–	10 / 10
Kabyle (kaby1243_Latin)	32.01	58.15	Afro-Asiatic	1	9 / 10
Kamba (Kenya) (kamb1297_Latin)	24.93	47.68	Atlantic-Congo	0	8 / 10
Kannada (nucl1305_Kannada)	55.88	63.90	Dravidian	1	10 / 10
Kashmiri (kash1277_Arabic)	26.62	62.82	Indo-European	1	10 / 10
Kashmiri (kash1277_Devanagari (Nagari))	22.55	57.73	Indo-European	1	10 / 10
Kazakh (kaza1248_Cyrillic)	64.98	72.01	Turkic	3	10 / 10
Kikuyu (kiku1240_Latin)	5.62	53.15	Atlantic-Congo	1	10 / 10
Kimbundu (kimb1241_Latin)	21.37	41.79	Atlantic-Congo	0	8 / 10
Kinshasa Lingala (ling1263_Latin)	48.36	53.12	Atlantic-Congo	1	10 / 10
Kinyarwanda (kiny1244_Latin)	59.06	65.75	Atlantic-Congo	1	10 / 10
Kirghiz (kirg1245_Cyrillic)	54.92	59.06	Turkic	1	10 / 10
Korean (kore1280_Hangul (Hangŭl, Hangeul))	38.21	62.31	Koreanic	4	10 / 10
Lao (laoo1244_Lao)	58.59	70.02	Tai-Kadai	2	10 / 10
Levantine Arabic (nort3139_Arabic)	67.19	74.01	Afro-Asiatic	–	10 / 10
Ligurian (ligu1248_Latin)	48.14	76.98	Indo-European	1	10 / 10
Limburgan (limb1263_Latin)	56.83	76.72	Indo-European	–	10 / 10
Lithuanian (lith1251_Latin)	65.99	71.04	Indo-European	3	10 / 10
Lombard (lomb1257_Latin)	40.32	67.99	Indo-European	1	10 / 10
Luba-Lulua (luba1249_Latin)	29.97	52.74	Atlantic-Congo	0	9 / 10
Luo (Kenya and Tanzania) (luok1236_Latin)	37.90	47.98	Nilotic	–	10 / 10
Macedonian (mace1250_Cyrillic)	64.95	70.12	Indo-European	1	10 / 10
Magahi (maga1260_Devanagari (Nagari))	57.93	73.59	Indo-European	0	10 / 10
Maithili (mait1250_Devanagari (Nagari))	50.43	66.99	Indo-European	1	10 / 10
Malayalam (mala1464_Malayalam)	59.07	69.10	Dravidian	1	10 / 10
Maltese (malt1254_Latin)	76.21	82.70	Afro-Asiatic	2	10 / 10
Mandarin (mand1415_Han (Simplified))	40.77	66.44	Sino-Tibetan	5	10 / 10
Mandarin (mand1415_Han (Traditional))	34.25	68.81	Sino-Tibetan	5	10 / 10
Manipuri (mani1292_Bengali)	19.06	64.31	Sino-Tibetan	0	10 / 10
Maori (maor1246_Latin)	47.45	64.97	Austronesian	1	10 / 10
Marathi (mara1378_Devanagari (Nagari))	52.66	66.06	Indo-European	2	10 / 10
Minangkabau (mina1268_Arabic)	8.12	61.44	Austronesian	1	10 / 10
Minangkabau (mina1268_Latin)	62.69	71.41	Austronesian	1	10 / 10
Mizo (lush1249_Latin)	50.39	59.40	Sino-Tibetan	0	10 / 10
Modern Greek (mode1248_Greek)	59.10	73.07	Indo-European	3	10 / 10
Modern Hebrew (hebr1245_Hebrew)	69.28	74.57	Afro-Asiatic	3	10 / 10
Moroccan Arabic (moro1292_Arabic)	45.14	60.62	Afro-Asiatic	5	10 / 10
Moselle Franconian (luxe1241_Latin)	59.83	75.58	Indo-European	1	10 / 10
Mossi (moss1236_Latin)	15.53	40.71	Atlantic-Congo	0	6 / 10
Najdi Arabic (najd1235_Arabic)	65.27	72.14	Afro-Asiatic	–	10 / 10
Nepali (nepa1254_Devanagari (Nagari))	52.28	70.34	Indo-European	1	10 / 10
North Azerbaijani (nort2697_Latin)	46.62	61.61	Turkic	–	10 / 10
Northern Kurdish (nort2641_Latin)	46.16	64.78	Indo-European	0	10 / 10
Northern Tosk Albanian (tosk1239_Latin)	64.32	74.24	Indo-European	–	10 / 10
Northern Uzbek (nort2690_Latin)	64.70	70.08	Turkic	–	10 / 10
Norwegian Bokmål (norw1259_Latin)	67.89	70.38	Indo-European	–	10 / 10
Norwegian Nynorsk (norw1262_Latin)	68.94	77.57	Indo-European	–	10 / 10

Continued on next page

Table 7 – continued from previous page

Language	$E \rightarrow T$	$T \rightarrow E$	Family	Class	Samples
(glottocode_Script)	chrF	chrF			($E \rightarrow T$ / $T \rightarrow E$)
Nuer (nuer1246_Latin)	6.65	21.75	Nilotic	0	4 / 8
Nyanja (nyan1308_Latin)	57.28	64.36	Atlantic-Congo	1	10 / 10
Occitan (occi1239_Latin)	64.46	75.99	Indo-European	1	10 / 10
Odia (oriy1255_Oriya)	57.08	70.09	Indo-European	1	10 / 10
Pangasinan (pang1290_Latin)	50.29	67.22	Austronesian	1	10 / 10
Papiamento (papi1253_Latin)	59.40	77.99	Indo-European	1	10 / 10
Pedi (pedi1238_Latin)	58.90	72.17	Atlantic-Congo	–	10 / 10
Plateau Malagasy (plat1254_Latin)	54.33	66.30	Austronesian	1	10 / 10
Polish (poli1260_Latin)	63.24	68.08	Indo-European	4	10 / 10
Portuguese (port1283_Latin)	74.12	72.46	Indo-European	4	10 / 10
Romanian (roma1327_Latin)	72.24	73.24	Indo-European	3	10 / 10
Rundi (rund1242_Latin)	46.17	59.44	Atlantic-Congo	1	10 / 10
Russian (russ1263_Cyrillic)	70.87	69.89	Indo-European	4	10 / 10
Samoan (samo1305_Latin)	52.34	70.75	Austronesian	1	10 / 10
Sango (sang1328_Latin)	18.31	41.16	Atlantic-Congo	1	8 / 10
Sanskrit (sans1269_Devanagari (Nagari))	38.77	53.26	Indo-European	2	10 / 10
Santali (sant1410_Ol Chiki (Ol Cemet’, Ol, Santali))	28.85	57.77	Austroasiatic	1	10 / 10
Sardinian (sard1257_Latin)	63.26	76.16	Indo-European	1	10 / 10
Scottish Gaelic (scot1245_Latin)	56.12	68.45	Indo-European	1	10 / 10
Serbian Standard (serb1264_Cyrillic)	63.79	74.85	Indo-European	4	10 / 10
Shan (shan1277_Myanmar (Burmese))	18.45	65.01	Tai-Kadai	0	6 / 10
Shona (shon1251_Latin)	50.02	53.16	Atlantic-Congo	1	10 / 10
Sicilian (sici1248_Latin)	50.63	68.78	Indo-European	1	10 / 10
Silesian (sile1253_Latin)	52.44	75.23	Indo-European	1	10 / 10
Sindhi (sind1272_Arabic)	56.57	71.45	Indo-European	1	10 / 10
Sinhala (sinh1246_Sinhala)	54.76	65.09	Indo-European	1	10 / 10
Slovak (slov1269_Latin)	59.60	68.26	Indo-European	3	10 / 10
Slovenian (slov1268_Latin)	70.90	72.76	Indo-European	3	10 / 10
Somali (soma1255_Latin)	48.80	62.48	Afro-Asiatic	1	10 / 10
South Azerbaijani (sout2697_Arabic)	37.49	63.69	Turkic	–	10 / 10
South Levantine Arabic (sout3123_Arabic)	53.99	70.58	Afro-Asiatic	–	10 / 10
South-Central Koongo (koon1244_Latin)	24.58	49.15	Atlantic-Congo	1	8 / 10
Southern Jinghpaw (kach1280_Latin)	21.18	45.03	Sino-Tibetan	0	6 / 10
Southern Pashto (sout2649_Arabic)	33.63	64.12	Indo-European	–	10 / 10
Southern Sotho (sout2807_Latin)	55.44	75.96	Atlantic-Congo	1	10 / 10
Southwestern Dinka (sout2832_Latin)	1.38	24.26	Nilotic	–	4 / 9
Spanish (stan1288_Latin)	63.33	66.93	Indo-European	5	10 / 10
Standard Arabic (stan1318_Arabic)	67.19	71.83	Afro-Asiatic	5	10 / 10
Standard Arabic (stan1318_Latin)	19.46	68.76	Afro-Asiatic	5	10 / 10
Standard Indonesian (indo1316_Latin)	74.66	69.53	Austronesian	3	10 / 10
Standard Latvian (stan1325_Latin)	63.66	73.36	Indo-European	3	10 / 10
Standard Malay (stan1306_Latin)	73.67	74.33	Austronesian	3	10 / 10
Sundanese (sund1252_Latin)	53.08	60.76	Austronesian	1	10 / 10
Swahili (swah1253_Latin)	75.19	77.87	Atlantic-Congo	2	10 / 10
Swati (swat1243_Latin)	47.46	59.26	Atlantic-Congo	1	10 / 10
Swedish (swed1254_Latin)	75.76	73.53	Indo-European	4	10 / 10
Ta’izzi-Adeni Arabic (taiz1242_Arabic)	57.90	68.61	Afro-Asiatic	–	10 / 10
Tagalog (taga1270_Latin)	65.38	79.03	Austronesian	3	10 / 10
Tajik (taji1245_Cyrillic)	57.78	65.02	Indo-European	1	10 / 10
Tamasheq (tama1365_Latin)	12.08	35.07	Afro-Asiatic	0	6 / 10
Tamasheq (tama1365_Tifinagh (Berber))	12.61	28.51	Afro-Asiatic	0	7 / 9
Tamil (tami1289_Tamil)	66.37	67.33	Dravidian	3	10 / 10
Tatar (tata1255_Cyrillic)	63.39	65.85	Turkic	1	10 / 10
Telugu (telu1262_Telugu)	58.70	74.29	Dravidian	1	10 / 10
Thai (thai1261_Thai)	64.09	74.75	Tai-Kadai	3	10 / 10
Tibetan (tibe1272_Tibetan)	46.95	58.32	Sino-Tibetan	1	10 / 10
Tigrinya (tigr1271_Ethiopic (Ge‘ez))	26.43	61.36	Afro-Asiatic	2	10 / 10
Tok Pisin (tokp1240_Latin)	46.00	58.89	Indo-European	1	10 / 10
Tsonga (tson1249_Latin)	53.82	66.83	Atlantic-Congo	1	10 / 10
Tswana (tswa1253_Latin)	45.34	62.99	Atlantic-Congo	2	10 / 10
Tumbuka (tumb1250_Latin)	48.32	58.45	Atlantic-Congo	1	10 / 10
Tunisian Arabic (tuni1259_Arabic)	43.91	67.20	Afro-Asiatic	–	10 / 10
Turkish (nucl1301_Latin)	69.30	78.82	Turkic	4	10 / 10
Turkmen (turk1304_Latin)	54.86	67.57	Turkic	1	10 / 10

Continued on next page

Table 7 – continued from previous page

Language **(glottocode_Script)**	$E \rightarrow T$ chrF	$T \rightarrow E$ chrF	**Family**	**Class**	**Samples** ($E \rightarrow T$ / $T \rightarrow E$)
Twi (twii1234_Latin)	40.08	54.68	Atlantic-Congo	1	10 / 10
Uighur (uigh1240_Arabic)	57.10	63.85	Turkic	1	10 / 10
Ukrainian (ukra1253_Cyrillic)	67.63	73.64	Indo-European	3	10 / 10
Umbundu (umbu1257_Latin)	19.95	44.89	Atlantic-Congo	0	7 / 10
Urdu (urdu1245_Arabic)	56.80	69.39	Indo-European	3	10 / 10
Venetian (vene1258_Latin)	53.60	72.88	Indo-European	1	10 / 10
Vietnamese (viet1252_Latin)	68.50	67.29	Austroasiatic	4	10 / 10
Waray (Philippines) (wara1300_Latin)	61.97	80.62	Austronesian	1	10 / 10
Welsh (wels1247_Latin)	76.84	80.79	Indo-European	1	10 / 10
West Central Oromo (west2721_Latin)	43.92	58.33	Afro-Asiatic	–	10 / 10
Western Farsi (west2369_Arabic)	51.22	69.55	Indo-European	–	10 / 10
Wolof (nucl1347_Latin)	27.23	52.05	Atlantic-Congo	2	9 / 10
Xhosa (xhos1239_Latin)	51.60	64.15	Atlantic-Congo	2	10 / 10
Yoruba (yoru1245_Latin)	25.90	50.06	Atlantic-Congo	2	10 / 10
Yue Chinese (yuec1235_Han (Traditional))	30.09	68.45	Sino-Tibetan	1	10 / 10
Zulu (zulu1248_Latin)	58.61	74.58	Atlantic-Congo	2	10 / 10

L Supplementary Figures

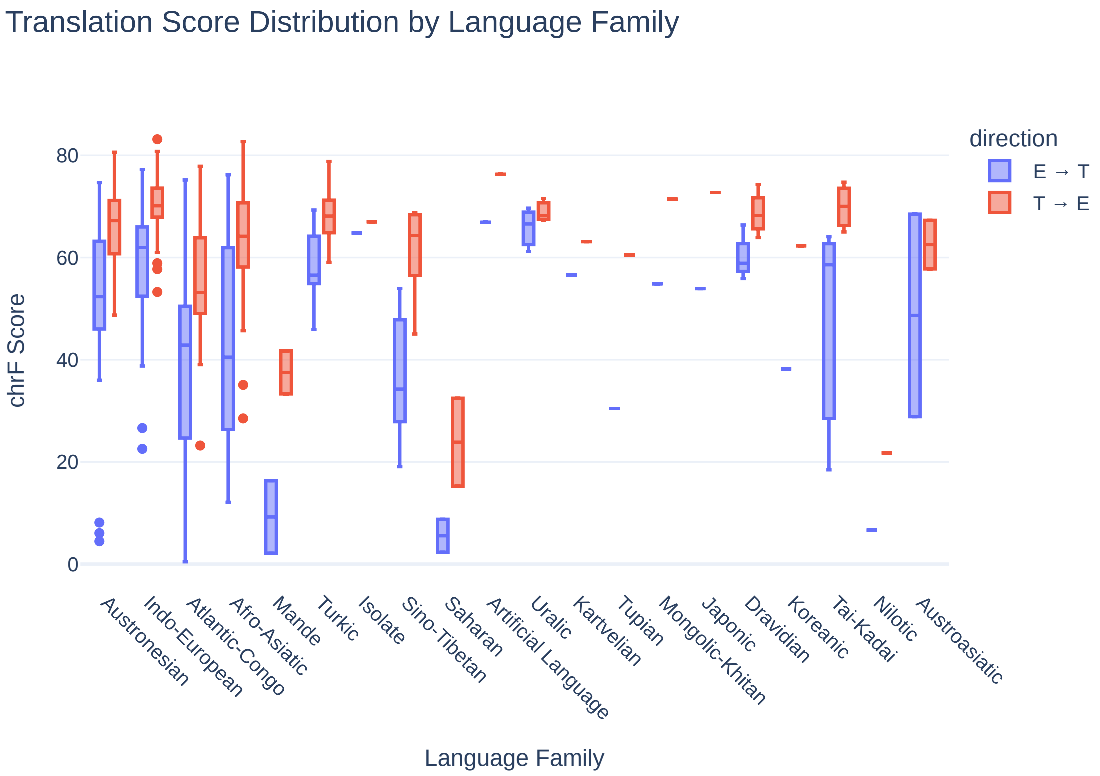


Figure 13: **Translation Score Distribution by Language Family.** This plot compares the distribution of chrF scores for English-to-Target ($E \rightarrow T$) and Target-to-English ($T \rightarrow E$) directions across language families. A consistent performance gap is evident, with $T \rightarrow E$ scores being almost universally higher and often less variable than $E \rightarrow T$ scores. Families such as Saharan and Mande show particularly low performance in the $E \rightarrow T$ direction, whereas families like Indo-European show a wider range of performance with generally higher scores.

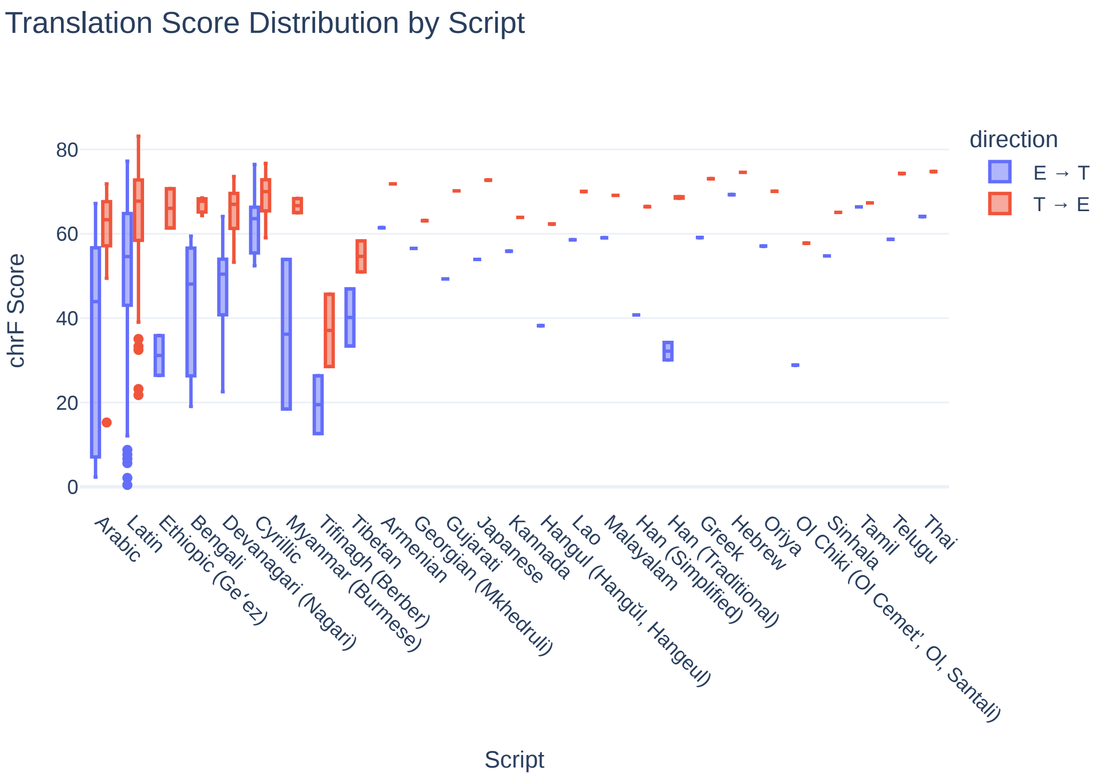


Figure 14: **Translation Score Distribution by Script.** This plot compares chrF score distributions across different writing systems. As with the family-based plot, the $T \rightarrow E$ direction consistently outperforms the $E \rightarrow T$ direction. Performance for languages using Latin and Cyrillic scripts is relatively high but shows a wide distribution, reflecting the diverse range of languages using them. Scripts associated with lower-resource languages, such as Ethiopic and Tifinagh, exhibit lower median scores, particularly in the $E \rightarrow T$ direction.

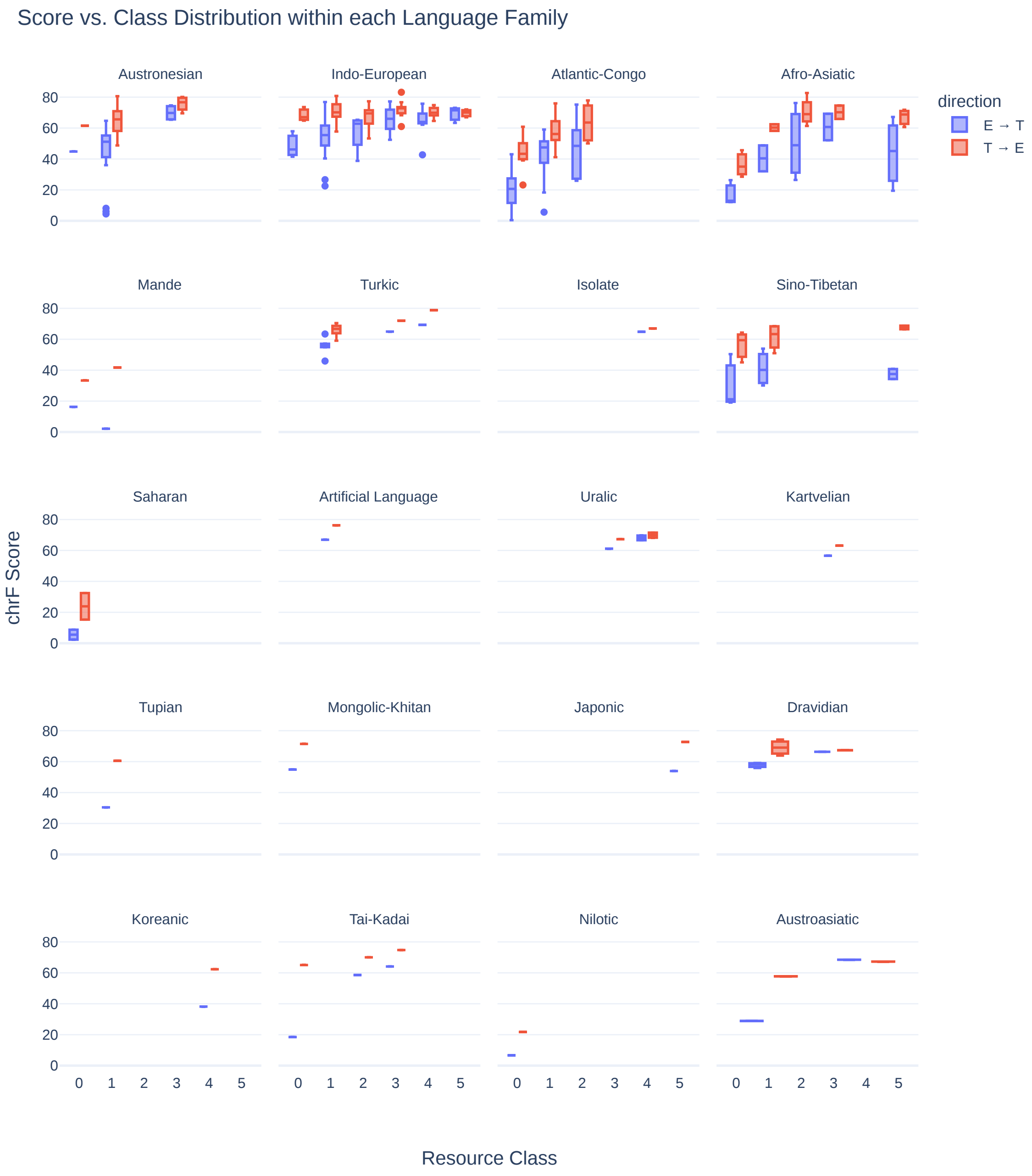


Figure 15: **Score vs. Class Distribution within each Language Family.** This faceted plot details the relationship between resource class and chrF score for each language family individually. A positive trend, where higher scores are associated with higher resource classes, is visible within several major families like Indo-European and Afro-Asiatic. The plot also highlights data sparsity, as many families (e.g., Mande, Saharan, Nilotic) contain languages in only one or two resource classes. The performance gap between the two translation directions persists even when controlling for class within a family.

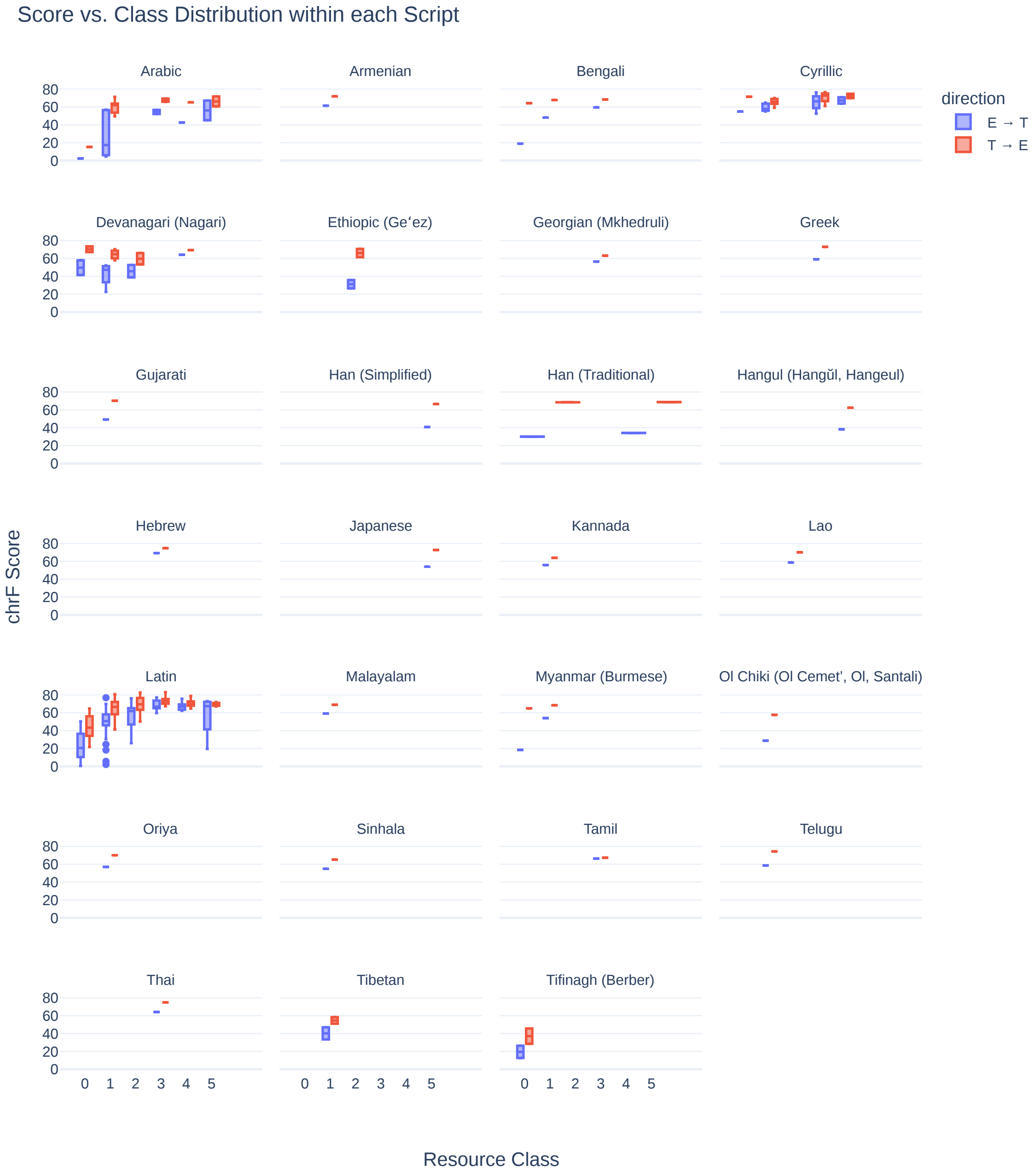


Figure 16: **Score vs. Class Distribution within each Script.** This faceted plot shows the relationship between resource class and chrF score for each writing system. The Latin script subplot contains the most data across all resource classes and most clearly demonstrates the positive correlation between class and score. For many other scripts, such as Arabic and Devanagari, the data is concentrated in the lower resource classes. This visualization confirms that the relationship between script and score is highly confounded with resource availability.

M Full Table of Model Performances

Run ID	**Avg Score (Answer)**	**Avg Score (Explanation)**	**Avg Score (Total)**	**p-value (Total)**
Gemini-2.5-pro (baseline)	0.385	0.520	0.443	N/A
OpenAI-o4-mini (baseline)	0.193	0.332	0.256	N/A
GPT-5 (baseline)	0.332	0.532	0.420	6.75×10^{-19}
Gemini-2.5-pro (guided)	0.392	0.537	0.454	2.11×10^{-1}
OpenAI-o4-mini (guided)	0.181	0.339	0.250	4.04×10^{-1}
Gemini-2.5-pro (w/ grammar agent)	0.383	0.533	0.448	5.50×10^{-1}
Gemini-2.5-pro (Single agent, 1^{st} round)†	0.383	0.522	0.444	N/A
Gemini-2.5-pro (Single agent, 2 rounds)	0.392	0.554	0.463	1.31×10^{-2}
Gemini-2.5-pro (Single agent, 3 rounds)	0.397	0.553	0.465	7.37×10^{-3}
Gemini-2.5-pro (Single agent, 4 rounds)	0.404	0.563	0.473	4.48×10^{-4}
Gemini-2.5-pro (Single agent, 5 rounds)	0.407	0.569	0.478	7.08×10^{-5}
Gemini-2.5-pro (Single agent, 6 rounds)	0.409	0.567	0.478	1.02×10^{-4}
OpenAI-o4-mini (Single agent, 1^{st} round)†	0.180	0.344	0.253	N/A
OpenAI-o4-mini (Single agent, 2 rounds)	0.191	0.357	0.264	2.40×10^{-1}
OpenAI-o4-mini (Single agent, 3 rounds)	0.192	0.367	0.269	6.69×10^{-2}
OpenAI-o4-mini (Single agent, 4 rounds)	0.197	0.357	0.267	1.30×10^{-1}
OpenAI-o4-mini (Single agent, 5 rounds)	0.199	0.371	0.274	1.20×10^{-2}
OpenAI-o4-mini (Single agent, 6 rounds)	0.198	0.378	0.276	4.29×10^{-3}
Gemini-2.5-pro (MoA, 1^{st} round)†	0.389	0.540	0.453	N/A
Gemini-2.5-pro (MoA, R=0, (2 rounds))	0.398	0.556	0.466	1.49×10^{-2}
Gemini-2.5-pro (MoA, R=1, (3 rounds))	0.410	0.573	0.480	7.74×10^{-5}
Gemini-2.5-pro (MoA, R=2, (4 rounds))	0.417	0.569	0.481	1.08×10^{-4}
Gemini-2.5-pro (MoA, R=3, (5 rounds))	0.418	0.581	0.488	1.06×10^{-5}
Gemini-2.5-pro (MoA, R=4, (6 rounds))	0.421	0.579	0.489	1.50×10^{-5}
OpenAI-o4-mini (MoA, first round)†	0.187	0.344	0.257	N/A
OpenAI-o4-mini (MoA, R=0 (2 rounds))	0.325	0.491	0.397	2.70×10^{-16}
OpenAI-o4-mini (MoA, R=1 (3 rounds))	0.359	0.513	0.427	2.83×10^{-18}
OpenAI-o4-mini (MoA, R=2 (4 rounds))	0.366	0.531	0.438	2.12×10^{-20}
OpenAI-o4-mini (MoA, R=3 (5 rounds))	0.384	0.537	0.451	1.83×10^{-20}
OpenAI-o4-mini (MoA, R=4 (6 rounds))	0.392	0.543	0.457	1.07×10^{-20}

Table 8: Summary of agent performance, showing average scores of “answer”, “explanation” and the combined total score. Each row represents a unique experimental setting. For the results with multiple rounds, the name denotes the model used in the final layer (i.e, the final solution is generated by it). The p-value is calculated with paired Student’s t-test, comparing the model with the baseline model of the same family. The rows marked with a dagger (†) means that its setting is equivalent to the baseline, and therefore the score differences demonstrate model stochasticity.

N Scores Categorized by Language Family and Problem Type

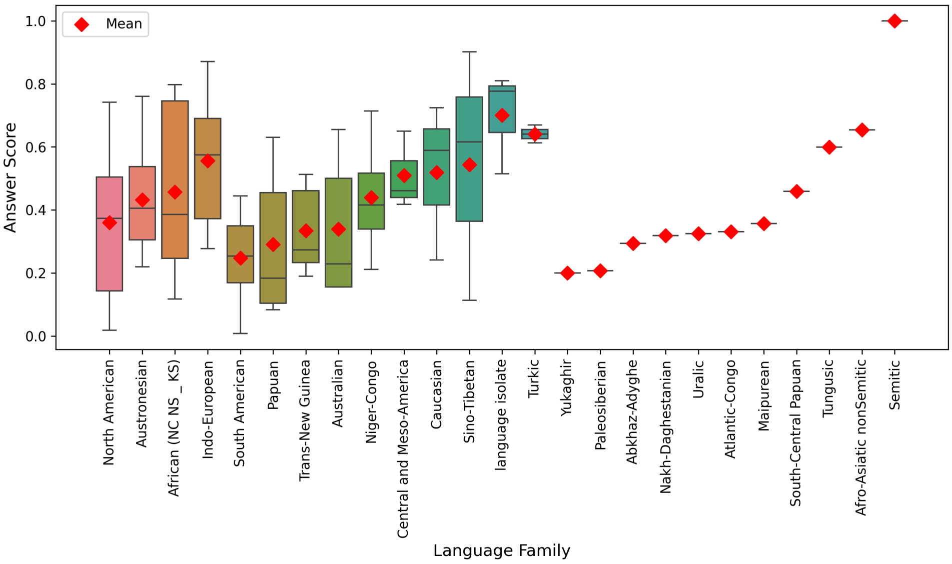


Figure 17: Distribution of Scores by Language Family.

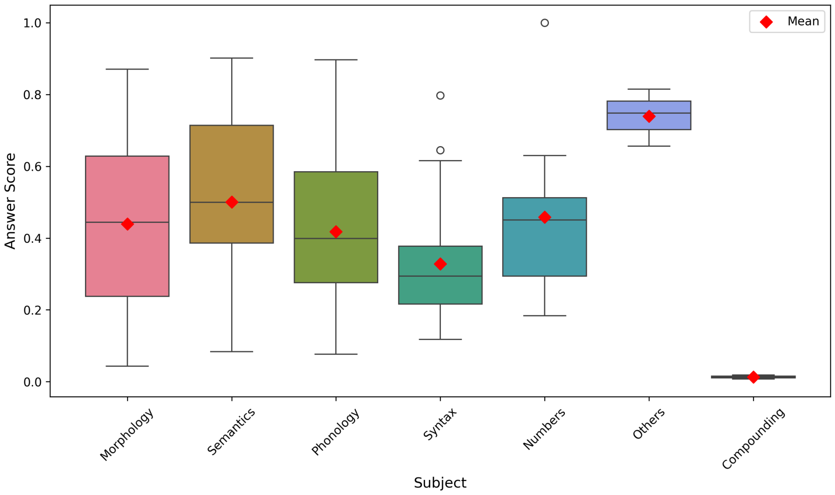


Figure 18: Distribution of Scores by Subject.

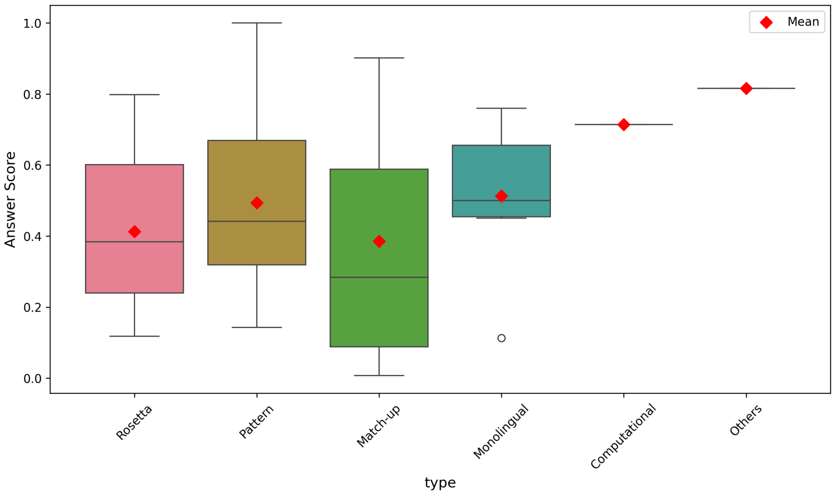


Figure 19: Distribution of Scores by Problem Type.

O Correlation between Answer Scores and Explanation Scores

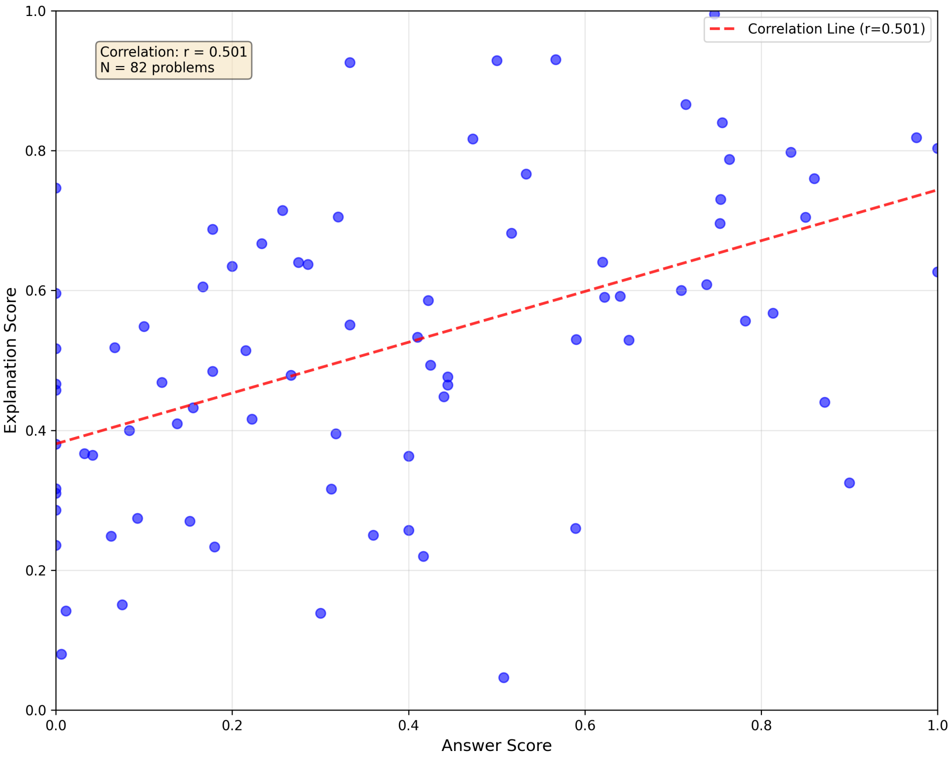


Figure 20: Correlation between Answer Scores and Explanation Scores.