

UNVEILING: What Makes Linguistics Olympiad Puzzles Tricky for LLMs?

Mukund Choudhary^{1,*†}, KV Aditya Srivatsa^{1,*}, Gaurja Aeron²,
Antara Raaghavi Bhattacharya^{3,#}, Dang Khoa Dang Dinh^{4,#}, Ikhlasul Akmal Hanif^{5,#},
Daria Kotova¹, Ekaterina Kochmar¹, Monojit Choudhury¹

¹Mohamed Bin Zayed University of Artificial Intelligence ²IIT Gandhinagar

³Harvard University ⁴VinUniversity ⁵Universitas Indonesia

Abstract

Large language models (LLMs) have demonstrated potential in reasoning tasks, but their performance on linguistics puzzles remains consistently poor. These puzzles, often derived from Linguistics Olympiad (LO) contests, provide a minimal contamination environment to assess LLMs’ linguistic reasoning abilities across low-resource languages. In this work, we analyze LLMs’ performance on 629 problems across 41 low-resource languages by labelling each with linguistically informed features to unveil weaknesses. Our analyses show that LLMs struggle with puzzles involving higher morphological complexity and perform better on puzzles involving linguistic features that are also found in English. We also show that splitting words into morphemes as a pre-processing step improves solvability, indicating a need for more informed and language-specific tokenisers. These findings thus offer insights into some challenges in linguistic reasoning and modelling of low-resource languages.

1 Introduction

In-context Data	
English	Bangime
The person hits me	yiwerε myū dara
We hear you	nne aw ne/no
We hit the bird	nne dεrε ne/dara
The fish drinks the water	ηε Vyε nnye
I see the bone	myū nεrε shura
Problems	Answers
We hear the bird	nne dεrε ne/no
ηε nεrε shura	The fish sees the bone

Colors are marked by annotators for better solvability and readability
/ = annotated morpheme boundary that was not a part of the original puzzle

ft	cf	eng (sim)	Broad Class
sov	0	0 (f)	Syntax
persfix	1	0 (f)	Morphology

Annotated LO Puzzle Breakdown

Figure 1: An LO puzzle with UNVEILING annotations. In-context data are translation pairs, and problems are to be translated on the basis of the data above.

to them as features akin to linguistic features (Dryer & Haspelmath, 2013). Fig. 1 shows an

Human proficiency in analysing languages and linguistic reasoning contributes significantly to efficient learning from limited linguistic input. Following frameworks and goals established by Mahowald et al. (2024); McClelland et al. (2019); Marcus (2020); Beguš et al. (2023), improvements in LLM metalinguistic processing capabilities can enhance general performance across diverse tasks. Linguistics Olympiad (LO) puzzles provide us with rich data to probe into these LLM abilities, especially in extremely low resource languages (LRL), as past research has shown (Chi et al., 2024). Typically, LO puzzles have parallel data between an unknown language and corresponding forms in a language known to the solver (Bozhanov & Derzhanski, 2013). The solver (typically pre-collegiate level) is expected to deduce the mapping between the linguistic structures of the unknown and the known languages, which involves observing syntactic patterns, semantic correspondences, phonological rules, or a combination of these. We shall refer

*Equal contribution. †Correspondence to mukund.choudhary@mbzuai.ac.ae.

#Work done during an internship at MBZUAI.

Dataset	Source	Puzzles (Problems)	Classes	Difficulty	Languages
MODELING (Chi et al., 2024)	Author-made	48 (272)	Noun-Adjective (40%), Semantics (10%), Word Order (40%), Possession (10%)	1 (3%), 2 (34%), 3 (23%), 4 (11%), 5 (29%)	19
PUZZLING (Şahin et al., 2020)	National Linguistics Olympiads	96 (752)	–	–	81
LINGOLY (Bean et al., 2024)	UK Linguistics Olympiad	90 (1133)	Compounding, Morphology, Numbers, Phonology, Semantics, Syntax	Breakthrough (6.5%), Foundation (13.3%), Intermediate (13.5%), Advanced (32.9%), Round 2 (33.7%)	94
LINGUINI (Sánchez et al., 2024)	International Linguistics Olympiad	160 (894)	Sequence Transduction, Fill-In Blanks, Number Transliteration	–	75

Table 1: Summary of datasets with Olympiad style puzzles. % = proportion in data

example of a puzzle set in Bangime (an unknown language) with the corresponding English (a known language) translations. The puzzles are similar to in-context few-shot learning tasks (Brown et al., 2020) where *context* contains all the data needed to solve the *problems*.

Generally, these are built from data in endangered languages or LRLs to ensure that participants do not already know the patterns in the data. Therefore, we expect these languages to be absent from the LLM training data as well. This makes LO puzzles a challenging and minimal contamination setting to assess an LLM’s linguistic reasoning abilities, and, for that reason, they have been incorporated into various benchmarks (Kazemi et al., 2025).

Several researchers have used LO puzzles to study language models’ reasoning abilities (summarized in Tab. 1). LLMs (while being impressive in application-based language tasks) perform consistently poorly in these Olympiads, whereas high school students can solve most of the puzzles fully. While these studies show that LLMs struggle, they do not explain why. Since these puzzles reflect the complex interplay of linguistic features in natural languages, pinpointing the causes of poor performance is challenging. On the other hand, a feature-level analysis of LLM weaknesses can enhance our understanding of their linguistic reasoning and can provide insights into bridging the gap between LLM abilities and a core component of human cognitive flexibility to identify patterns, extract rules, and generalise from minimal data across unfamiliar languages.

To answer the research question of *what linguistic features make LO puzzles challenging for LLMs*, we have annotated 64 puzzles (629 problems, 41 LRLs) from existing datasets (Sec. 2.1) with 50 linguistic features (from WALS,¹ also including other attributes such as similarity to English – see Sec. 2.2). We discover that LLM performance (Sec. 2.3) is significantly negatively correlated with the number of morphological features of a language present in an LO puzzle (Sec. 3.1) and the number of features in a puzzle that have a low coverage of exemplifying data in the puzzle (Sec. 3.3). LLM scores also positively correlated with the average number of features in a puzzle that are present in English (Sec. 3.2). Further experiments showed that breaking down words in LRL into morphemes in morphologically complex puzzles helped LLMs (Sec. 3.5), seemingly indicating the need for more language-specific handling of tokenisation of words in LRL. Findings from the study could inform LLM-based research on sample-efficient LRL learning, metalinguistic reasoning, and more.

2 Setting the Stage for UNVEILING

In this section, we set the stage for unveiling LLM weaknesses in solving LO puzzles by:

1. Curating data and finding unseen puzzles.
2. Establishing an annotation scheme to break down puzzles into linguistic features.
3. Obtaining LLM responses to evaluate performance against the feature annotations.

2.1 Data

Existing Datasets An overview of the four datasets that are based on LO-style puzzles can be found in Tab. 1. All of these reveal that LLMs are poor at solving LO problems based

¹World Atlas of Languages by Dryer & Haspelmath (2013)

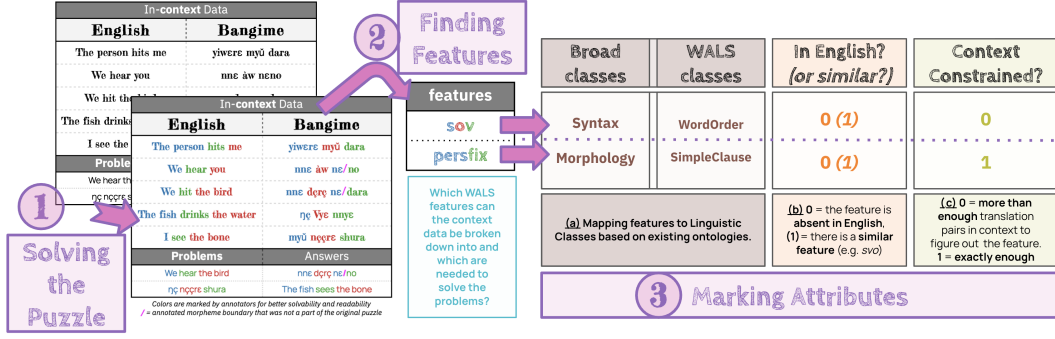


Figure 2: UNVEILING annotation pipeline and attributes marked in a sample from the data.

on Exact-Match Accuracy, which measures the percentage of problems per puzzle where the predicted final answer exactly matches the ground truth. LLMs performed the best (40-60%) in solving puzzles from MODELING (Chi et al., 2024) (author-made, easier LO-style puzzles) and Foundation-level (younger students) puzzles in LINGOLY (Bean et al., 2024), while performing the worst (25%) on International Linguistics Olympiad (IOL) puzzles from LINGUINI (Sánchez et al., 2024).

Filtering out Solvable Puzzles LLMs struggle with LINGUINI’s set of IOL puzzles as they are tougher and are designed for participants who have more experience with LO puzzles. Thus we only consider puzzles from MODELING (M) and LINGOLY (L). To maintain uniformity across datasets and select the simplest tasks, we retained only Rosetta-Stone style (Bozhanov & Derzhanski, 2013) puzzles which had paired translations in context and problems that required participants to either translate from an LRL to English or vice versa. This filtering step led to an initial pool of puzzles consisting of all 48 M puzzles² and 45 L puzzles out of 90.

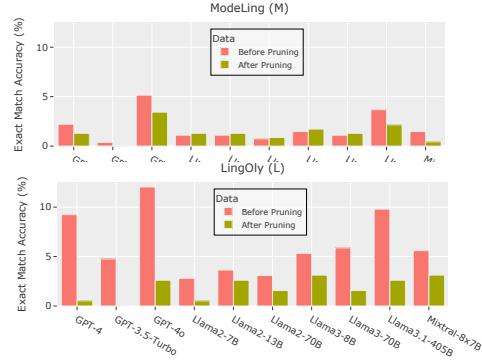


Figure 3: Results on the NoContext prompt to find puzzles with potential contamination.

The final dataset contains 64 puzzles (629 problems):⁴ 41 (272) M + 23 (357) L.⁵

Contamination Checks and Pruning MODELING (M) and LINGOLY (L) employ measures to account for data contamination, i.e., being privy to the exact puzzles or being familiar with the languages involved, which may lead to inflated results. M puzzles are based on extremely low-resource languages that were reported to have 0% data leakage to OpenAI models up to GPT-4. Bean et al. (2024) (L) report all analyses with a NoContext baseline for querying problems without the puzzle in-context data and improvements over it.

We employ a similar measure by using a **NoContext** prompt (App. A) followed by a manual inspection of all model responses and drop the contaminated puzzles from both datasets (Fig. 3): 7 out of 48 M puzzles (answered by the GPT-4o³ without any in-context data) and 22 out of 45 L puzzles were pruned. This is expected as there are more high-resourced languages (e.g., Romani and Nepali) in the L puzzles set, and corresponding puzzles are available online via UKLO website.

The final dataset contains 64 puzzles (629 problems):⁴ 41 (272) M + 23 (357) L.⁵

²All M puzzles were first solved and checked by the authors for inconsistencies (L puzzles were already checked by UKLO participants). Issues included insufficient data, absence of affixes in answers, etc. A detailed account of minor changes made to the data is provided in App. B.

³This model was not public at the time of MODELING’s publication.

⁴Each puzzle is made of multiple translation problems, which are evaluated independently.

⁵Remaining contamination ($\leq 4\%$) was where models matched answers by picking random words and piecing together a possible solution from both languages’ context and questions.

2.2 Feature & Attribute Annotations

Background & Motivation LLM performance explanations have been limited to using broad external difficulty metrics (e.g. participant scores or Likert ratings – see Tab. 1). As LLMs might model language differently (McCoy et al., 2024; Warstadt et al., 2020), such measures conflate linguistic reasoning with puzzle-specific quirks and offer low interpretability.

A more comprehensive probe of LLM linguistic abilities in English was done by Warstadt et al. (2020) and Waldis et al. (2024). They craft linguistic textbook-style tasks to find that linguistic competence correlates with the model size, but also that different features have different learning curves and instruction tuning negatively influences performance and natural quality – i.e., they mimic linguistic features instead of generalising productively like a native speaker. They also find that formal linguistic abstractions like *syntax* in English are more easily modelled by LLMs as compared to less explicitly lexically or structurally represented abstractions like *pragmatics*.

LO puzzles offer an opportunity to conduct similar studies on LLM abilities in LRLs. However, as they are based on real language (not artificially controlled) data, each puzzle involves simultaneously changing features from different linguistic abstractions like *phonology*, and *semantics* (Fig. 1).

Thus, the following sections describe how we analyse LO puzzles by building on top of Bozhanov & Derzhanski (2013)’s insight that LO puzzles are crafted “to be solved by deducing linguistic patterns, with other sources of technical complexity ... not making them linguistically interesting”. As Fig. 2 indicates, we break them down into fine-grained, linguistically grounded proxies for complexity, which we call “*Attributes*” from this point in the paper. Attributes include 50 linguistic “*Features*” from abstractions like *morphology*, *syntax*, etc., and other derived attributes such as how many features are similar to English features, and if there are redundant context pairs.

Attribute	Classes	μ	σ
Proportion of Broad Attributes per Puzzle	b_Syntax	0.519	0.381
	b_Morphology	0.307	0.322
	b_Semantics	0.132	0.243
	b_Phonology	0.042	0.11
Proportion of WALS Attributes per Puzzle	w_WordOrder	0.437	0.399
	w_ComplexSentences	0.004	0.023
	w_NominalSyntax	0.096	0.186
	w_SimpleClauses	0.131	0.192
	w_VerbalCategories	0.03	0.078
	w_NominalCategories	0.164	0.244
	w_Morphology	0.075	0.181
	w_Lexicon	0.023	0.131
	w_Phonology	0.039	0.109
Proportion of Data Constrained Features per Puzzle (cf)		0.474	0.435
Proportion of Features also Present in English per Puzzle (eng)		0.298	0.36
Proportion of Features Similar to English per Puzzle (eng_sim)		0.498	0.381

Table 2: Summary Statistics of all attributes – means (μ) and standard deviations (σ).

Methodology 2 ex-LO participants – with training in Linguistics and LO puzzle design – independently annotated all puzzles with WALS linguistic features and classified them in three ways (outlined below). For an overview of the annotation pipeline, see Fig. 2; the annotation guidelines are presented in App. F).

2.2.1 Solving Puzzles

All puzzles were first solved independently by the annotators. They also marked the context data used to translate the problems to help with the next steps.

2.2.2 Finding Features (ft)

Broken-down contexts were then used to extract features. A feature here (defined by Dryer & Haspelmath (2013), similar to Warstadt et al. (2020)’s *phenomenon*) is a structural property of language that describes one aspect of cross-linguistic diversity. Annotations were based on a set of 50 WALS features (App. D.) comprising various grammatical and semantic affixation, vowel harmony, abstract concepts like moiety, and more. Feature description with examples and citations can be found in App. D.⁶

⁶Code and annotations are publicly available at <https://github.com/mukundc2k/unveiling>

2.2.3 Marking for Attributes

(a) Linguistic Attributes We cluster features into linguistic abstractions defined by WALS and UKLO⁷ for a more generalized but still linguistically grounded analysis of LLM abilities. Based on UKLO, we define **Broad classes (b.)**: Syntax (phrasal/clausal rules), Phonology (sound units), Morphology (subword units), Semantics (meaning units), and based on WALS, we define **WALS classes (w.)**:

- Nominal/VerbalCategories: Relating to nouns and verbs such as case and tense.
- WordOrder: Order of sentence constituents like adjective-noun or subject-object.
- Phonology: Features based on units of sound, tones, vowel harmony, and so on.
- Lexicon: Lexical items representing number systems, alienability, and so on.
- ComplexSentences: Relative clauses and other multi-clause constructions.
- NominalSyntax/SimpleClauses: Minor parts like possessive or voice markers.
- Morphology: Reduplication and other features that use morphemes.

(b) Similarity to English (eng): To unveil biases via typological distance from English, we marked no. of features that were present in English or showed similar behavior (sim).

(c) Constrained features (cf): It is generally easier for a solver to uncover a feature and its rules in an LO puzzle when it appears in different contexts with redundant minimal pairs of translations. A puzzle is tougher if another feature changes in the same evidence pairs. Thus to unveil if LLMs face similar difficulties, we mark how many features have exactly enough translation pairs to uncover the respective feature from the context data.⁸

2.2.4 Resolving Annotation Differences

Finally, after comparing the independent annotations, we found that the annotators agreed almost perfectly on which WALS features were present in a puzzle (Cohen’s $\kappa=0.893$), substantially on how redundancies per puzzle ($\kappa=0.631$), and on the count of features per puzzle that were present in English as well ($\kappa=0.761$).

They resolved the disagreements by relabeling the puzzles with newly agreed-upon reasoning. Specifically, for finding constrained features, they independently came up with pairs in contexts relevant to uncovering the feature and agreed on a union. For English similarity, they referred to WALS and other research in English linguistics (cited for all features in App. D).

Tab. 2 provides a statistical summary of all attributes used for analyses (Sec. 3).

2.3 Benchmarking Experiments

We evaluate 10 LLMs:⁹ GPT 4, 4o, 3.5-Turbo (OpenAI, 2024), Llama2 7B, 13B, 70B (Touvron et al., 2023), Llama3 8B, 70B, 3.1-405B (Dubey et al., 2024), and Mixtral 7x8B (Jiang et al., 2024). These were prompted with 6 styles and strategies from previous work.¹⁰

⁷A quick reference for WALS classes and for UKLO classes

⁸There can never be fewer than enough contexts by design.

⁹Models were selected to be representative of various parameter sizes, families, and to contain both open & closed models for balance and comparability. Refer to App. A for more details.

¹⁰M marked prompts are borrowed from Chi et al. (2024), L is borrowed from Bean et al. (2024).

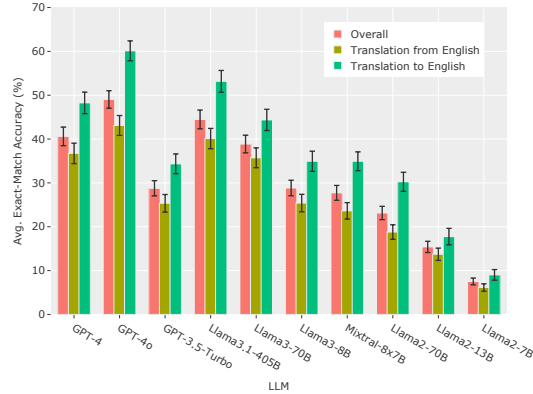


Figure 4: Benchmarking results – overall and language direction-specific avg. Exact-Match accuracies for all models across prompts.

- **Null**: prompt to gauge model performance without explicit instructions.
- **Minimal^M**: similar to Null but with task description.
- **Hand-Tuned^M**: prompt fine-tuned by an IOL medalist.
- **Basic CoT^M**: Chain-of-Thought prompt asking the model to answer step-by-step.
- **Full CoT^M**: similar to above but with an example of reasoning in a dummy puzzle.
- **LingOly^L**: A more detailed prompt similar to the Minimal prompt above.

We find that our results on the curated dataset are in line with existing literature (Fig. 4):

- Larger and closed-source models are better (Srivastava et al., 2023). Smaller models (Mixtral-8x7B and Llama3-8B) outperform larger ones (GPT3.5-Turbo) rarely, which can be explained by Waldis et al. (2024)’s finding that linguistic skills vary depending on architecture and tuning parameters.
- Models have a translation direction bias and score better when translating from the low-resource language *to* English as compared to *from* (Şahin et al., 2020).

For a more detailed breakdown and other observations, please refer to App. C. Note that we use a model’s average performance across prompts for the following analyses.

3 Analyses UNVEIL that:

To unveil patterns behind LLM performances we compute the correlations between the attributes and exact match scores across prompts. Analyses are only based on significant correlations.

3.1 Morphologically complex puzzles are difficult for LLMs

We find that the count of morphological features (b.Morphology) in an LO puzzle strongly predicts how well a model can solve the puzzle.

Tab. 3¹¹ shows that higher morphological feature count correlates with lower exact match scores across all LLMs (all significant p-values $\leq .001$). These scores are averaged over all prompting strategies; however, this observation holds over most individual model-prompt combinations as well. For all combinations, please refer to Tab. 12 in Appendix.

To provide some illustrative examples, a puzzle in Ayutla Mixe (Mixe-Zoque family, Fig. 5) encoded many morphological features, and no LLMs solved more than one question. In contrast, Bangime (language isolate from Mali, Fig. 1) had only one such feature (affix for person), most LLMs solved all the problems except for the one that involved the affix. Finally, a puzzle with no morphological complexity, based on Engenni (Niger-Congo family, Fig. 7), was solved fully by all models.

Note that smaller and older models that, on average, score lower across all puzzles also showed lower correlations with b.Morphology. Accurate answers by these models were mostly random or chance matches and were not consistent across questions in the same puzzle (better scoring models would either consistently not apply a morphological rule or use it productively across answers).

In-Context Translation Pairs	Ējts nexp → I see
	Mejts mtunp → You work
	Juan yē'ē yē'exeyejtpy → Juan watches him
	yē'ē yē'uk yexpy → He sees the dog
	Ējts yē' maxu'unk nexyejtpy → I watch the baby
Question-Answer Pairs	1sg. 2sg. 3sg. noun (obj.) verbs +for long verbal affix
	yē' maxu'unk yexp → The baby sees
	Mejts Juan mtunpy → You work (for) Juan
	I work (for) him → Ējts yē'ē ntunpy
	The baby watches the dog → yē' maxu'unk yē'uk yexyejtpy

Figure 5: An annotated Ayutla puzzle. Colors mark equivalent units.

Mapudungan	
kūmé wentru → good man	
kūmé pafu → good turkey	
::	::
Mixtepec Zapotec	
guits nquits → white paper	
mĒEd nquits → white child	
::	::

Figure 6: Sample of context from two puzzles.

¹¹"*", "***", and "****" imply that corresponding p-value is less than 0.05, 0.01, and 0.001, respectively.

	Llama2 (7B)	Llama2 (13B)	Llama2 (70B)	Mixtral (8x7B)	Llama3 (8B)	Llama3 (70B)	Llama3.1 (405B)	GPT (3.5-Turbo)	GPT (4o)	GPT (4)
Average Exact-Match Score	0.074	0.143	0.224	0.271	0.269	0.381	0.405	0.248	0.488	0.408
Count of b.Morphology	-0.219***	-0.348***	-0.496***	-0.546***	-0.522***	-0.583***	-0.572***	-0.483***	-0.637***	-0.553***
Average of eng	0.031	0.053	0.388**	0.3*	0.491***	0.364**	0.276*	0.399**	0.258*	0.381**
Count of cf	-0.144	-0.198	-0.448***	-0.466***	-0.467***	-0.55***	-0.592***	-0.543***	-0.572***	-0.547***

Table 3: Correlation values (Pearson) between exact-match scores and attribute value aggregates per puzzle for selected attributes against all models across prompts.

In Sec. 3.5, we actively test our hypothesis of LLMs being weak in splitting morphemes into the right tokens in morphologically richer languages. Results show that explicit morpheme boundaries increase the scores in better-performing models.

3.2 LLMs have an English bias

We find that a puzzle’s average number of features that are present in English (avg_eng) strongly predicts how well an LLM scores on that puzzle. Tab. 3 shows that a higher average similarity positively correlates with higher exact match scores. Note that the average model-prompt combinations have a slightly lower significance level ($p\text{-value} \leq .01$); however, many individual model-prompt combinations are consistent with this observation (Tab. 12).

As illustrative examples (Fig. 6), a puzzle in Mixtepec Zapotec (Oto-Manguean family) required the solver to figure out Noun-Adjective order (opposite to English); however, LLMs could not solve more than 30% of the questions in this puzzle. In contrast, Mapudungan (Araucanian family) had the same setup as Mixtepec Zapotec, except that the word order in noun-adjective phrases in this language is the same as in English, so most LLMs fully solved this puzzle.

Note that most correlations are in the same range and consistent across prompts and models, all model types are similarly biased towards English feature-like puzzles.

3.3 More constrained evidence for a feature makes a puzzle difficult for LLMs

We find that the count of features with exactly enough pairs needed to uncover them (cf) in an LO puzzle strongly predicts how well a model can solve the puzzle. Tab. 3 shows that higher exact feature count correlates with lower exact match scores across most LLMs (all significant $p\text{-value} \leq .001$ and negative correlations). Note that these scores are averaged over different prompting strategies however, this observation holds over most individual model-prompt combinations as well (Tab. 12).

The Engenni puzzle (Fig. 7) shows that the last two examples in the puzzle are redundant and are not required to uncover either the feature (Subject-Verb order) or vocabulary. Thus, there is more than enough evidence to figure out all features. LLMs scored higher on such puzzles – i.e., puzzles where both annotators found ample evidence for features in them.

Note that smaller and older models lack these correlations due to low performance in general (especially on more data-constrained puzzles). Further, Null and Minimal prompts (which did not explain the task) weaken correlations for other mid-sized models.

3.4 Other Observations

Some other statistics and significant correlations that were only localised to certain model prompt combinations are summarised below.¹²

¹²Other attributes that were not mentioned in the above analyses were highly correlated with the features that were not represented in more than half of the dataset, to make any generalised claims.



Figure 7: Engenni Annotated

w.WordOrder showed significant and negative correlations with the smaller and older models (Llama2-7B & 13B) for specific prompts. Note that these are also the models that perform poorly overall and show no correlation with other attributes. *Semantics*- and *phonology*-based attributes across the two classification systems were scarce in the dataset, leading to low significance scores. However, at a slightly lower significance level ($p\text{-value} \leq .01$), the models from OpenAI showed a negative correlation, implying that puzzles with more *semantics*- or *phonology*-based features were tougher for OpenAI models (consistent with Waldis et al. (2024); Warstadt et al. (2020)). We also extended our experimental results to DeepSeek-R1, a model tuned for Inference-Time Compute (DeepSeek-AI et al., 2025) – see Appendix C.1 for more details.

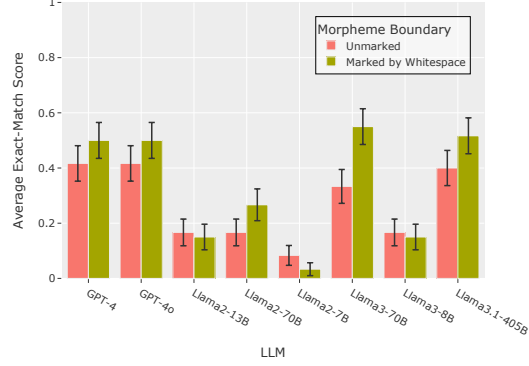


Figure 8: Performance change in models upon making morpheme boundaries explicit

3.5 Breaking morphemes down helps LLMs

Insightful findings from Sec. 3.1 unveiled that the morphological complexity of an LO puzzle strongly predicted LLM scores. We further probe a subset of 9 puzzles of varying morphological complexity and feature types on which models did not perform well.

As illustrated with pink bars in Fig. 1 and with boxes in Fig. 5, we first annotate the morpheme boundaries in the context (based on grammar (App. D) and answers), problems, and answer data of these puzzles.

Then we run inference using selected models (with prompt styles that led to best scores respectively) again, with two variations of morpheme boundary representations: whitespace-separated (which made morpheme boundaries the same as the word boundaries), and @-separated. We observe that there is a substantial increase in scores, both quantitatively and qualitatively (Fig. 8 and Fig. 20). Note that we maintain the original tokenizer for all experiments and add separation characters (i.e., whitespace or @) to the input text before tokenization. Although there is overlap between the resultant word-piece tokens and true morphemes, it is incidental and low (with GPT-4, we observed only a 0.62 Jaccard similarity between the morphemes and the word-pieces).

Puzzle	Exact-Match Score (+Gain)
<i>L_Ilokano</i>	0.333 (+0.333)
<i>L_Karelian</i>	0.769 (+0.154)
<i>L_Lardil</i>	0.333 (+0.167)
<i>M_Ayutla_Mixe</i>	0.0 (+0.0)
<i>M_Bangime_3</i>	0.4 (+0.2)
<i>M_Bangime_5</i>	0.6 (+0.2)
<i>M_Guugu_Yimithir</i>	0.7 (+0.2)
<i>M_Kutenai</i>	1.0 (+0.6)
<i>M_Totonac</i>	0.333 (+0.167)

Table 4: Llama3-70B puzzle-wise score gain on marking morpheme boundaries explicitly

The results thus show that all models that were performing above 15% on the subset solved more problems accurately after splitting. While Llama2-7B, 13B, and Llama3-8B were mostly unaffected by the splits, other models like Llama3-70B (Tab. 4) reached 50% accuracy, solving some puzzles better than the baseline scores of much bigger models. The substantial score improvements after morphological splitting demonstrate that morphology was indeed the primary bottleneck, while the remaining performance gaps in some puzzles reveal the contribution of non-linguistic reasoning components.

Furthermore, separating morpheme boundaries by a whitespace or by ‘@’ did not lead to very different results. Fig. 9 shows Llama3-70B’s response change after split on an extended version of the Bangime puzzle from Fig. 1. The response was formatted and worded before and after splitting in exactly the same way by the LLM (we have added ellipses for simpler presentation), except that after marking the morpheme boundaries, it identified the right morphemes and their translations after splitting.

Finally, we also find that LLMs could not solve some puzzles with morphemes that packed a lot of semantics (e.g., kinship in Lardil) or with a lot of morphemes (e.g., Ayutla Mixe – Fig. 5).

However, for most cases (Tab. 11), we observe that just splitting LRL context data into morphemes can lead to better translations.

4 Discussion & Related Work

As interest grows in understanding how LLMs’ processing of language aligns with linguistic theory (Futrell & Mahowald, 2025), researchers find that LM abilities pertain to specific linguistic features rather than languages in the context of multilingual tasks (Acs et al., 2024). The UNVEILING method probes LLMs in a similar fashion. We find that breaking LO puzzles down into linguistic attributes to study LLMs is reliable (i.e., it confirms some existing findings from an intrinsic perspective of linguistic features in the puzzle instead of Likert-based difficulty ratings (Chi et al., 2024)) and can be expanded (i.e., it allows us to unpack more about LLM performance).

Data: We show that data constraint adds to a puzzle’s difficulty for LLMs by annotating redundancy for each feature in a puzzle. Other research shows extrinsically that puzzles are made easier for LLMs by adding natural language data across linguistic abstractions (Zhu et al., 2025), prompting models to generate data from a typologically close language (Ramji & Ramji, 2024), or by augmenting data through character replacements (Watanabe, 2024).¹³

English bias: We show a feature-level typological bias towards linguistic features in English. Past research in LM performance on LO puzzles and results from Sec. 2.3 arrive at this conclusion extrinsically by showing that there is also translation directionality bias towards English, where all models showed that LLMs are better at translating into English as compared to translating into an LRL. Outside of LO puzzles, in a cross-linguistic generalization ability probing of LLMs through a Wug Test (Anh et al., 2024), authors found that most LLMs do not add affixes to nonce words, add them incorrectly, or even use English affixes in another lower-resourced language and context.

Interestingly, Warstadt et al. (2020) find LMs to have concentrated robust knowledge of specific domains of grammar in English (based on supervised training on acceptability data of minimal pairs). They note that “these studies cover a limited set of phenomena, excluding ellipsis, quantification, and countless others. This is likely due to the labour-intensive nature of collecting such targeted minimal pairs”. We believe that LO puzzles are by design positioned as a neat set of minimal pairs in extremely low-resourced languages, addressing this problem.

Morphology & Tokenisation: While our analyses show a significant correlation between poorer LLM performance and an increasing number of morphological features in an LO puzzle (based on an unseen natural language), most other LO research confirms this very superficially. Upon making morpheme boundaries explicit, we found that LLMs performed better generally. This implies a need for a deeper focus on revamping tokenisation approaches by either researching ways for tokenisers to adapt to language-specific contexts or using tokeniser-free approaches.

Much like our experiments, Ismayilzada et al. (2024) investigate the morph generalization abilities of LLMs through the lens of compositionality and find a major gap in their perfor-

```

Let's analyze the given examples to find patterns and
relationships between the Bangime phrases and their
English translations.

From the examples, we can observe the following:

* "myú" seems to be related to "I" ...
* "nne" appears to be related to "we"
- * "neno" contains "no", ...
+ * "ne" is a suffix that appears in phrases with
  "we" or "I" ...

...
- * "nejiná" is associated with "swim".
+ * "jiná" means "swim".
* "dçrç" is associated with "the bird".

Now, let's try to translate .. "The fish swims".

From the examples, we know that
"gc" is .. "the fish",
- and "nejiná" is .. "swim".
+ and "jiná" means "swim".

Therefore, a logical combination would be:
- English: "The fish swims"
Bangime: "gc nejiná"
+ Bangime: "gc jiná"

```

Figure 9: Llama3-70B response change (before to after) recognises the person affix after splitting morphemes and reaches the correct final answer.

¹³For most of the work cited here, the data creation strategy is either not fully explained, or synthetic data is either linguistically infeasible or not validated by native speakers.

mance as compared to humans (in agglutinative languages). [Weissweiler et al. \(2023\)](#) also apply a Wug test in typologically diverse languages (English, German, Tamil, and Turkish) to arrive at the same conclusion.

[Haslett \(2025\)](#) finds that this is because LLMs do not directly perceive units like affixes and semantic radicals. Here, subword tokens are determined by frequency, not semantics, and end up obscuring meaningful units. Larger vocabularies of longer units hide sub-morphemic units from LLMs.

However, the tokenisation problem has been recently proven to be NP-complete ([Whittington et al., 2024](#)), and various new solutions, such as Greedy tokenisation ([Lim et al., 2025](#)), Byte-free tokenisation (where dedicated local transformers identify dynamically sized patches of bytes) ([Pagnoni et al., 2024](#)), or trigram-based hashing of vocabulary ([Deiseroth et al., 2024](#)), might help with the morphology gap.

Another line of relevant work is to use language-specific tokenisers. [Dang et al. \(2024\)](#); [Bayram et al. \(2025\)](#) find that subword-based and morphologically informed tokenization works better for Turkish, whereas character-level tokenization benefits Hindi morphology. New methods like MorphBPE (BPE with an additional step of blocking pair-merging apart from morpheme boundaries) ([Asgari et al., 2025](#); [Lerner & Yvon, 2025](#)) showed faster convergence in training and more alignment with morphologies of languages with different richness ([Fanar et al., 2025](#)).

Consequently, there is a need for more intrinsic approaches to evaluation of model abilities in morphological processing. [Ali et al. \(2024\)](#) show that low fertility scores are a necessary but not sufficient criterion for this. [Arnett & Bergen \(2024\)](#) also claim that byte-premiums and scaling data need to be factored in with morphology-tokenisation alignment differences to explain performance degradation in LRLs. Based on our analyses of LLMs, we believe that designing translation puzzles in unseen languages (like LO puzzles), controlled for specific and rare morphological features, is a starting point for that.

5 Conclusions & Future Work

We presented a linguistic feature-based breakdown of Linguistics Olympiad style puzzles and a correlation study between various features of linguistics puzzles and the performance of LLMs on these puzzles. Our analyses show that LLMs are better at solving puzzles with more English-like features, struggle to solve puzzles with more features that have constrained data to uncover them, and puzzles that have a higher number of morphological features in low-resource and unseen languages. This shows that LLMs can be studied in more depth in terms of their linguistic reasoning and puzzle-solving abilities and weaknesses in a Linguistics Olympiad-style setting.

We further probed models by making morpheme boundaries explicit to unveil that LLMs performed better with this small change in the puzzles. This motivates a deeper look into tokenisation techniques used in multi-lingual settings, specifically for extremely low-resource languages or unseen data.

We believe that a linguistic feature-based analysis of LO puzzles can provide a better insight into how LLMs unpack new language data, and that we could design more targeted LO-style puzzles to unveil specific aspects. Other analyses could include addressing different kinds of morphology, e.g. inflectional as well as derivational, to better understand the observed effects. We could also explore how different prompting strategies change LLM responses and their consistency in solving LO puzzles ([Lin et al., 2023](#)).

Finally, we hope that this work may also inform other tasks such as studying multi-modal LLM capabilities in deciphering low-resource languages' orthographies ([Shih et al., 2025](#)) or in making educational language learning apps ([Vasselli et al., 2024](#)) annotated with hand-crafted features to bolster LLM abilities in modelling low resource languages.

Ethics Statement

With regard to the datasets used in our work, we comply with all available licenses and/or have sought authors’ explicit permission to use their datasets. We have made minimal changes and will publish only mappable annotations and a list of fixes for full reproducibility. The annotators volunteered for the task and are co-authors of the paper. We foresee no other major ethical concerns that could arise from this work or the data annotated.

Reproducibility Statement

All code and data are publicly released at <https://github.com/mukundc2k/unveiling>, ensuring that our experiments can be independently reproduced. To guarantee consistency, all experiments involving LLMs were performed with a temperature setting of 0. Detailed information about the specific model variants used and the complete querying specifications can be found in App. A. We also note that the contamination check performed in this study does not rule out the possibility that models released after the study may have been trained on the publicly available subset of puzzles used herein.

References

- Judit Acs, Endre Hamerlik, Roy Schwartz, Noah A Smith, and Andras Kornai. Morphosyntactic probing of multilingual BERT models. *Natural Language Engineering*, 30(4):753–792, 2024.
- Mehdi Ali, Michael Fromm, Klaudia Thellmann, Richard Rutmann, Max Lübbering, Johannes Leveling, Katrin Klug, Jan Ebert, Niclas Doll, Jasper Buschhoff, et al. Tokenizer choice for LLM training: Negligible or crucial? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 3907–3924, 2024.
- Dang Anh, Limor Raviv, and Lukas Galke. Morphology Matters: Probing the Cross-linguistic Morphological Generalization Abilities of Large Language Models through a Wug Test. In Tatsuki Kuribayashi, Giulia Rambelli, Ece Takmaz, Philipp Wicke, and Yohei Oseki (eds.), *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pp. 177–188, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.cmcl-1.15. URL <https://aclanthology.org/2024.cmcl-1.15/>.
- Catherine Arnett and Benjamin K Bergen. Why do language models perform worse for morphologically complex languages? *arXiv preprint arXiv:2411.14198*, 2024.
- Ehsaneddin Asgari, Yassine El Kheir, and Mohammad Ali Sadraei Javaheri. MorphBPE: A Morpho-Aware Tokenizer Bridging Linguistic Complexity for Efficient LLM Training Across Morphologies. *arXiv preprint arXiv:2502.00894*, 2025.
- M Ali Bayram, Ali Arda Fincan, Ahmet Semih Gümüş, Sercan Karakaş, Banu Diri, and Savaş Yıldırım. Tokenization Standards for Linguistic Integrity: Turkish as a Benchmark. *arXiv preprint arXiv:2502.07057*, 2025.
- Andrew M Bean, Simi Hellsten, Harry Mayne, Jabez Magomere, Ethan A Chi, Ryan Chi, Scott A Hale, and Hannah Rose Kirk. LINGOLY: A Benchmark of Olympiad-Level Linguistic Reasoning Puzzles in Low-Resource and Extinct Languages. *arXiv preprint arXiv:2406.06196*, 2024.
- Gašper Beguš, Maksymilian Dabkowski, and Ryan Rhodes. Large linguistic models: Analyzing theoretical linguistic abilities of LLMs. *arXiv preprint arXiv:2305.00948*, 2023.
- Bozhidar Bozhanov and Ivan Derzhanski. Rosetta stone linguistic problems. In *Proceedings of the Fourth Workshop on Teaching NLP and CL*, pp. 1–8, 2013.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini

- Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20*, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.
- Nathan Chi, Teodor Malchev, Riley Kong, Ryan Chi, Lucas Huang, Ethan Chi, R. McCoy, and Dragomir Radev. MODELING: A Novel Dataset for Testing Linguistic Reasoning in Language Models. In Michael Hahn, Alexey Sorokin, Ritesh Kumar, Andreas Shcherbakov, Yulia Otmakhova, Jinrui Yang, Oleg Serikov, Priya Rani, Edoardo M. Ponti, Saliha Muradoğlu, Rena Gao, Ryan Cotterell, and Ekaterina Vylomova (eds.), *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pp. 113–119, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.sigtyp-1.14>.
- Thao Anh Dang, Limor Raviv, and Lukas Galke. Tokenization and Morphology in Multilingual Language Models: A Comparative Analysis of mT5 and ByT5. *arXiv preprint arXiv:2410.11627*, 2024.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanji Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning, 2025. URL <https://arxiv.org/abs/2501.12948>.
- Björn Deiseroth, Manuel Brack, Patrick Schramowski, Kristian Kersting, and Samuel Weinbach. T-FREE: Subword Tokenizer-Free Generative LLMs via Sparse Representations for Memory-Efficient Embeddings. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21829–21851, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1217. URL <https://aclanthology.org/2024.emnlp-main.1217/>.

- Matthew S. Dryer and Martin Haspelmath (eds.). *WALS Online (v2020.3)*. Zenodo, 2013. doi: 10.5281/zenodo.7385533. URL <https://doi.org/10.5281/zenodo.7385533>.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, and Angela Fan, et al. The Llama 3 Herd of Models, 2024. URL <https://arxiv.org/abs/2407.21783>.
- Team Fanar, Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, et al. Fanar: An Arabic-Centric Multimodal Generative AI Platform. *arXiv preprint arXiv:2501.13944*, 2025.
- Richard Futrell and Kyle Mahowald. How Linguistics Learned to Stop Worrying and Love the Language Models. *arXiv preprint arXiv:2501.17047*, 2025.
- David A Haslett. Tokenization changes meaning in large language models: Evidence from Chinese. *Computational Linguistics*, pp. 1–30, 2025.
- Mete Ismayilzada, Defne Circi, Jonne Sällevä, Hale Sirin, Abdullatif Köksal, Bhuwan Dhingra, Antoine Bosselut, Duygu Ataman, and Lonneke van der Plas. Evaluating morphological compositional generalization in large language models. *arXiv preprint arXiv:2410.12656*, 2024.
- Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Léo Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of Experts, 2024. URL <https://arxiv.org/abs/2401.04088>.
- Mehran Kazemi, Bahare Fatemi, Hritik Bansal, John Palowitch, Chrysovalantis Anastasiou, Sanket Vaibhav Mehta, Lalit K. Jain, Virginia Aglietti, Disha Jindal, Peter Chen, Nishanth Dikkala, Gladys Tyen, Xin Liu, Uri Shalit, Silvia Chiappa, Kate Olszewska, Yi Tay, Vinh Q. Tran, Quoc V. Le, and Orhan Firat. BIG-Bench Extra Hard. *arXiv preprint arXiv:2502.19187*, 2025.
- Paul Lerner and François Yvon. Unlike “Likely”, “Unlike” is Unlikely: BPE-based Segmentation hurts Morphological Derivations in LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5181–5190, 2025.
- Jia Peng Lim, Davin Choo, and Hady W Lauw. A partition cover approach to tokenization. *arXiv preprint arXiv:2501.06246*, 2025.
- Zheng-Lin Lin, Chiao-Han Yen, Jia-Cheng Xu, Deborah Watty, and Shu-Kai Hsieh. Solving Linguistic Olympiad Problems with Tree-of-Thought Prompting. In Jheng-Long Wu and Ming-Hsiang Su (eds.), *Proceedings of the 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)*, pp. 262–269, Taipei City, Taiwan, October 2023. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP). URL <https://aclanthology.org/2023.rocling-1.33>.
- Kyle Mahowald, Anna A Ivanova, Idan A Blank, Nancy Kanwisher, Joshua B Tenenbaum, and Evelina Fedorenko. Dissociating language and thought in large language models. *Trends in cognitive sciences*, 28(6):517–540, 2024.
- Gary Marcus. The next decade in AI: four steps towards robust artificial intelligence. *arXiv preprint arXiv:2002.06177*, 2020.
- James L McClelland, Felix Hill, Maja Rudolph, Jason Baldridge, and Hinrich Schütze. Extending machine language models toward human-level language understanding. *arXiv preprint arXiv:1912.05877*, 2019.

- R. Thomas McCoy, Shunyu Yao, Dan Friedman, Mathew D. Hardy, and Thomas L. Griffiths. Embers of autoregression show how large language models are shaped by the problem they are trained to solve. *Proceedings of the National Academy of Sciences*, 121(41): e2322420121, 2024. doi: 10.1073/pnas.2322420121. URL <https://www.pnas.org/doi/abs/10.1073/pnas.2322420121>.
- OpenAI. Hello GPT-4o, May 2024. URL <https://openai.com/index/hello-gpt-4o/>.
- Artidoro Pagnoni, Ram Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason Weston, Luke Zettlemoyer, et al. Byte latent transformer: Patches scale better than tokens. *arXiv preprint arXiv:2412.09871*, 2024.
- Raghav Ramji and Keshav Ramji. Inductive Linguistic Reasoning with Large Language Models. *arXiv preprint arXiv:2412.17819*, 2024.
- Gözde Gül Şahin, Yova Kementchedjheva, Phillip Rust, and Iryna Gurevych. PUZZLING Machines: A challenge on learning from small data. *arXiv preprint arXiv:2004.13161*, 2020.
- Eduardo Sánchez, Belen Alastruey, Christophe Ropers, Pontus Stenetorp, Mikel Artetxe, and Marta R Costa-jussà. LINGUINI: A benchmark for language-agnostic linguistic reasoning. *arXiv preprint arXiv:2409.12126*, 2024.
- Yu-Fei Shih, Zheng-Lin Lin, and Shu-Kai Hsieh. Reasoning Over the Glyphs: Evaluation of LLM’s Decipherment of Rare Scripts. *arXiv preprint arXiv:2501.17785*, 2025.
- Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, pp. 1–95, 2023. ISSN 2835-8856.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, and Shruti Bhosale, et al. Llama 2: Open Foundation and Fine-Tuned Chat Models, 2023. URL <https://arxiv.org/abs/2307.09288>.
- Justin Vasselli, Arturo Martínez Peguero, Junehan Sung, and Taro Watanabe. Applying linguistic expertise to LLMs for educational material development in indigenous languages. In *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP 2024)*, pp. 201–208, 2024.
- Andreas Waldis, Yotam Perlitz, Leshem Choshen, Yufang Hou, and Iryna Gurevych. Holmes: Benchmark the Linguistic Competence of Language Models. *arXiv preprint arXiv:2404.18923*, 2024.
- Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. BLiMP: The Benchmark of Linguistic Minimal Pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020. doi: 10.1162/tacl.a.00321. URL <https://aclanthology.org/2020.tacl-1.25/>.
- Junehan Sung Hidetaka Kamigaito Taro Watanabe. Exploring Metalinguistic Awareness in Pre-trained Language Models through the International Linguistics Olympiad Challenges. *anlp*, 2024.
- Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Hao-fei Yu, et al. Counting the Bugs in ChatGPT’s Wugs: A Multilingual Investigation into the Morphological Capabilities of a Large Language Model. *arXiv preprint arXiv:2310.15113*, 2023.
- Philip Whittington, Gregor Bachmann, and Tiago Pimentel. Tokenisation is NP-Complete. *arXiv preprint arXiv:2412.15210*, 2024.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. HuggingFace’s Transformers: State-of-the-art Natural Language Processing, 2020. URL <https://arxiv.org/abs/1910.03771>.

Hongpu Zhu, Yuqi Liang, Wenjing Xu, and Hongzhi Xu. Evaluating Large Language Models for In-Context Learning of Linguistic Patterns In Unseen Low Resource Languages. In *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, pp. 414–426, 2025.

A Querying Details

A.1 Models

Tab. 5 presents the corresponding model versions for the LLMs used in our experiments and analyses. All GPT-* models were accessed from a dedicated OpenAI API and all other models were imported from HuggingFace (Wolf et al., 2020) and loaded and inferred using vLLM.

A.2 Computational Resources

We query all models with a temperature of 0 and a max output token length of 1024 for each problem response. The exact prompt templates used for querying are as follows:

LLM	Model Version
GPT-4	gpt-4-0613
GPT-4o	gpt-4-turbo-2024-04-09
GPT-3.5-Turbo	gpt-3.5-turbo-0125
Llama3.1-405B	meta-llama/Llama-3.1-405B-Instruct
Llama3-70B	meta-llama/Meta-Llama-3-70B-Instruct
Llama3-8B	meta-llama/Meta-Llama-3-8B-Instruct
Llama2-70B	meta-llama/Llama-2-70b-chat-hf
Llama2-13B	meta-llama/Llama-2-13b-chat-hf
Llama2-7B	meta-llama/Llama-2-7b-chat-hf
Mixtral-7x8B	mistralai/Mixtral-8x7B-Instruct-v0.1

Table 5: Model versions for the LLMs used in our experiments and analyses.

A.3 Prompts

In this section, we present the various prompts used in our experiments and analyses.

1. Null Prompt (Only data from the puzzle and one problem at a time, without any explicit instructions.) – See Figure 10.
2. MODELING Minimal Prompt (Chi et al., 2024) – See Figure 11.
3. MODELING Hand-Tuned Prompt (Chi et al., 2024) – See Figure 12.
4. MODELING Basic CoT Prompt (Chi et al., 2024) – See Figure 13.
5. MODELING Full CoT Prompt (Chi et al., 2024) – See Figure 14.
6. LINGOLY Std Prompt (Bean et al., 2024) – See Figure 15.
7. No Context Prompt (Context pairs from data in the puzzle withheld for detecting contamination.) – See Figure 16.

In each of the below prompt templates, the span "<<DATA>>" corresponds to the translation pairs of the source language to the target language provided as data for a puzzle. This is presented in Figure 17. The exact layout of an individual problem ("<<PROBLEM>>") and all the problems ("<<ALL_PROBLEMS>>") in a puzzle are presented in Figure 18.

Null Prompt
=====

<<DATA>>

<<PROBLEM>>

Figure 10: Prompt template for Null prompt

Modeling Minimal Prompt
=====

Here are some expressions in <<LANG>> (a never-seen-before foreign language) and their translations in English:

<<DATA>>

Given the above examples, please translate the following expression.

<<PROBLEM>>

Figure 11: Prompt template for Modeling Minimal prompt

Modeling HandTuned Prompt
=====

This is a translation puzzle. Below are example phrases in <<LANG>> (a never-seen-before foreign language) as well as their English translations. Some test phrases follow them. Your task is to look closely at the example phrases and use only the information from them to translate the test phrases.

<<DATA>>

Given the above examples, please translate the following expression.

<<PROBLEM>>

Figure 12: Prompt template for Modeling HandTuned prompt

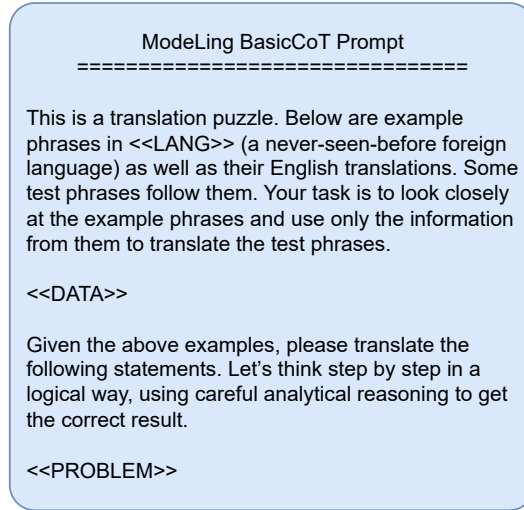


Figure 13: Prompt template for ModeLing BasicCot prompt

B MODELING corrections

Here we present a list of changes to the MODELING dataset that we made, along with our rationale. Each problem is referred to with a language name and a number if necessary.¹⁴

- **Ainu:** We removed “=” symbols: they are used to help beginner readers understand morphology.¹⁵ However, it is not clear, whether those symbols would help an LLM or confuse it.
- **Ayutla Mixe:** We changed “Ėjts yĕ’ĕ nexpy” in answers to “Ėjts yĕ’ĕ ntunpy” as a translation for “I work for him”. “Tun” is the root for “work”, while “ex” is the one for “see”. Similarly, we changed “Yĕ’ maxu’unk yĕ’ uk yexpy.” to “Yĕ’ maxu’unk yĕ’ uk yexyejtpy.” in the solutions for consistency.¹⁶
- **Bangime 4:** We changed “nnE dĉrĉ nEnonnE” in answers to “nnE dĉrĉ nEno” as translation to “we hear the bird”, because the “nnE” suffix is not present in the examples in the question or even in the reference.¹⁷
- **Bangime 7:** We changed “bad food” in the questions to “bad land” (answer: “gwyE begaũ”) because “bad food” is already in the given parallel sentences. We also changed “big house” to “big food” (answer: “jyEkE kanyoro”) to maintain the same number for each noun in the data and questions.
- **Dogon:** We changed “ú ıló” in solutions to “ú ılò” because the High-High (HH) pattern was not used anywhere in the examples (even in Dogon 2). The rule, as we understand it, is that the possessed noun tones are LL in the examples with non-pronominal noun possessors and HL in the pronominal noun possessors. This can be cross-checked in section 6.2.1.1 (pg. 141-143) of the reference.¹⁸
- **Kalam:** We changed the question “kapkap yib ag” to “kamiket yib ag” and the corresponding answer from “speak very slowly” to “speak very stealthily” as the word/root for “slowly” is not in the data.
- **Kutenai:** We replaced “stomach” in the questions with “belly” and “wum” in answers with “wumnana”. The original question requires inferring from “wumnana

¹⁴we use E in Bangime to represent the mid-open front vowel sound

¹⁵Ainu for beginners

¹⁶A reference grammar for Ayutla Mixe

¹⁷Bangime reference

¹⁸A Grammar of Toro Tegu (Dogon) Tabi mountain dialect

Modeling FullCoT Prompt
=====

This is a translation puzzle. In a moment, you will use logic and analytical reasoning to translate from a never-seen-before language <<LANG>> to English. As a training example, here are some expressions in Spanish and their translations in English.

1. Spanish: ventana roja
English: red window
2. Spanish: ventana azul
English: blue window
3. Spanish: manzana azul
English: blue apple

Using the above examples, translate the following.
Spanish: manzana roja

ANSWER: English: red apple

EXPLANATION: The first step we notice is that the word "ventana" must mean window because (1) the word "ventana" appears twice between sentences 1 and 2, and (2) the only word that appears twice in the English translation is "window." Next, we infer that "roja" must be "red" and "azul" must be "blue" by process of elimination. Next, we guess that in Spanish, the noun precedes the adjective because "ventana" comes before "roja" and "azul." Therefore, the noun in sentence 3 ("apple") must correspond to the word preceding the adjective ("manzana") in the Spanish translations. Putting this together, "manzana roja" must mean "red apple" in English.

Using a similar logical and analytical reasoning to understand the grammar of the foreign languages step by step, look closely at the following example in <<LANG>> phrases and use only the information from them to translate the following test phrases.

<<DATA>>

Given the above examples, please translate the following statements.

<<PROBLEM>>

Figure 14: Prompt template for Modeling FullCoT prompt

LingOly Std Prompt
=====

Below is a problem sheet from a linguistics exam.
You will first see the entire mapped data and
questions, then be asked to respond to the questions
individually. Your answers to the questions should
rely only on reasoning about the information
provided in the sheet.
Data:

<<DATA>>

Questions:

<<ALL_PROBLEMS>>

Now respond to the following question from the set:

<<PROBLEM>>

Figure 15: Prompt template for LingOly prompt

NoContext Prompt
=====

This is a translation puzzle from a linguistics exam.
Below are some test phrases to be translated in
<<LANG>> (a never-seen-before foreign language)
and some in English. You will be asked to translate a
specific phrase from the set.

<<ALL_PROBLEMS>>

Now translate the following phrase:

<<PROBLEM>>

Figure 16: Prompt template for NoContext prompt

```
// Context Pair #1
<<SRC_LANG>>: "<<SRC_TEXT>>"
<<TRG_LANG>>: "<<TRG_TEXT>>"

// Context Pair #2
<<SRC_LANG>>: "<<SRC_TEXT>>"
<<TRG_LANG>>: "<<TRG_TEXT>>"

...

// Context Pair #N
<<SRC_LANG>>: "<<SRC_TEXT>>"
<<TRG_LANG>>: "<<TRG_TEXT>>"
```

Figure 17: Prompt template for <<DATA>> span. Lines starting with "/" are comments only for reference and not part of the actual span.

Single Problem (<<PROBLEM>>)

```
<<SRC_LANG>>: "<<SRC_TEXT>>"
<<TRG_LANG>>:
```

All Problems in Puzzle (<<ALL_PROBLEMS>>)

```
// Problem #1
<<SRC_LANG>>: "<<SRC_TEXT>>"
<<TRG_LANG>>:

// Problem #2
<<SRC_LANG>>: "<<SRC_TEXT>>"
<<TRG_LANG>>:

...

// Problem #N
<<SRC_LANG>>: "<<SRC_TEXT>>"
<<TRG_LANG>>:
```

Figure 18: Prompt template for "<<PROBLEM>>" and "<<ALL_PROBLEMS>>" spans. Lines starting with "/" are comments only for reference and not part of the actual span.

- belly" that belly is a "small stomach", however official definitions state that "belly" is the general area of the stomach and adjuncts. In addition, "nana - small" is being tested in another question.

- **Mapudungan 1:** We changed "pichi mansun" to "pichi pafu" (answer: "small turkey") as the original phrase was translated already in the data.
- **Mixtepec Zapotec 2:** We changed "zhàb ròsâd" (pink clothing) to "yâg ròsâd" (pink stick) for the same reason as in the previous task.
- **Ngadha 1:** We changed "big leaf" in examples to "big fruit" as there is not enough data in the problem to establish that "fruit" ("li'e") and "leaf" are the same (if they are) in the answers.

C Benchmarking details

We report the average exact match accuracy of each model-prompt combination in Tab. 7 across our dataset (O = M+L) and across the M & L subsets.

Here is an overview of our benchmarking experiments, which show results in line with previous literature (Tab. 1 and BIG-Bench (Srivastava et al., 2023)), demonstrating that closed-source, larger models are generally better solvers, and puzzles set in low-resource languages are generally more difficult to solve.

Along Models and Prompts We find that GPT-4o performs the best, and while GPT-4 scores slightly lower, it shows more consistency across prompts than other models. In general, larger models within a model family perform better. However, for suitable prompts, relatively smaller models – Llama3-8B and Mixtral-8x7B – outperform larger models like Llama2-70B and GPT-3.5-Turbo. This aligns with Waldis et al. (2024)’s observation that, while parameter size generally correlates with linguistic competence, specific architecture and tuning parameters can impact models’ capabilities, particularly in the domains of morphology and syntax.

Consistent with the results of Sánchez et al. (2024), who find that OpenAI models work better with fewer shot prompting, we also find that GPT-4o and Mixtral-8x7B perform better with more concise prompts (i.e., Null and Minimal).

Model	Average Accuracy		t
	M	L	
GPT-4	56.1	13.5	12.55***
GPT-3.5-Turbo	35.1	6.6	9.99***
GPT-4o	65.7	18.6	15.77***
Llama2-7B	9.3	4.1	3.45**
Llama2-13B	19.4	5.1	6.33***
Llama2-70B	31.9	5.3	10.77***
Llama3-8B	38.4	6.6	11.05***
Llama3-70B	53.2	11.2	13.03***
Llama3.1-405B	55.3	14.1	11.93***
Mixtral-8x7B	39.1	5.8	12.13***

Table 6: Average accuracy on M puzzles is significantly higher than L puzzles for all models. *t*-statistics marked with ‘***’ indicates a *p*-value $\leq .001$.

Full CoT provides models with a descriptive solution to a short linguistic puzzle in Spanish on noun phrase construction and proceeds to instruct the models to solve the actual linguistic puzzle similarly. Thus, it is possible that many of the models perform well with the Full CoT prompt as it provides a short demonstration in a known language.

Along Dataset Sources All models score higher on M puzzles than on L puzzles. As described earlier, this is because M puzzles are made in the style of LO puzzles but involve a lower number of linguistic features per question and have more data per feature to uncover it in the puzzle.

Along Translation Direction Consistent with Şahin et al. (2020)’s findings for LLMs before 2021, we find that most models also score better when translating from the low-resource language *to* English compared to when translating *from* English (see Figure 19). This difference is most pronounced in GPT-4o’s results on M puzzles.

C.1 Extended Results for DeepSeek-R1

Table 8 presents the exact match accuracy on the entire evaluation set (O) for DeepSeek-R1, which has been tuned for Inference-Time Compute (DeepSeek-AI et al., 2025). We compare these scores with the best scores for each prompt setting as shown in Table 7. We observe that R1 demonstrates superior performance while maintaining our observed correlations between model performance and morphological features (Pearson correlation = -0.74, *p*-value < 0.001). To better determine the impact of ITC on this significant gain, future work must extend our study to a wider set of ITC models.

Model	Null Prompt			Modeling Minimal			Modeling Hand-Tuned			Modeling Basic CoT			Modeling Full CoT			LINGOLY Std			Model-Average		
	O	M	L	O	M	L	O	M	L	O	M	L	O	M	L	O	M	L	O	M	L
GPT-4	41.7	58.2	12.3	41.3	58.1	11.4	40.2	54.3	15.0	41.1	55.7	15.2	42.0	56.9	15.3	38.4	53.6	11.4	40.8	56.1	13.5
GPT-3.5-Turbo	5.2	6.4	3.1	25.8	37.6	4.8	33.4	47.3	8.6	25.3	34.6	8.5	32.8	47.2	7.3	26.5	37.4	7.1	24.8	35.1	6.6
GPT-4o	47.4	67.1	12.2	53.1	71.7	19.8	49.2	66.0	19.4	47.6	60.9	23.8	47.0	62.9	18.5	48.3	65.4	17.8	48.8	65.7	18.6
Llama2-7B	7.0	7.0	7.0	7.0	9.3	2.9	8.3	11.5	2.6	8.9	11.8	3.7	8.7	10.2	5.9	4.7	5.8	2.7	7.4	9.3	4.1
Llama2-13B	8.6	8.8	8.1	14.4	19.8	4.7	20.1	28.8	4.7	14.4	20.3	3.9	19.0	26.2	6.3	9.1	12.6	2.9	14.3	19.4	5.1
Llama2-70B	18.6	26.6	4.2	23.5	34.7	3.5	25.3	36.1	6.1	22.4	30.8	7.3	25.9	36.2	7.5	18.6	27.1	3.4	22.4	31.9	5.3
Llama3-8B	17.5	24.4	5.2	30.4	43.3	7.4	29.4	42.8	5.4	27.8	38.3	9.1	30.2	42.8	7.6	26.4	38.5	5.0	26.9	38.4	6.6
Llama3-70B	34.3	50.8	5.0	37.0	52.4	9.6	37.8	50.3	15.3	40.5	55.4	13.7	39.8	55.4	12.2	39.3	54.7	11.7	38.1	53.2	11.2
Llama3.1-405B	20.7	27.5	8.5	31.7	43.0	11.6	48.6	66.5	16.8	48.7	64.1	21.2	49.0	67.8	15.6	44.3	63.0	10.8	40.5	55.3	14.1
Mixtral-8x7B	23.9	35.5	3.3	32.5	45.3	9.8	31.8	46.0	6.5	24.0	34.6	5.2	23.1	33.0	5.3	27.2	39.9	4.6	27.1	39.1	5.8
Prompt Average	22.5	31.2	6.9	29.7	41.5	8.6	32.4	45.0	10.1	30.1	40.7	11.2	31.7	43.9	10.2	28.3	39.8	7.7	29.1	40.3	9.1

Table 7: Summary of an average exact match accuracy of each model-prompt combination across our dataset (O) and subsets of puzzles from MODELING (M) and LINGOLY (L). Models are divided by family and arranged by parameter size. **Bold** scores mark the best model for each prompt, underlined scores mark the best prompts per model.

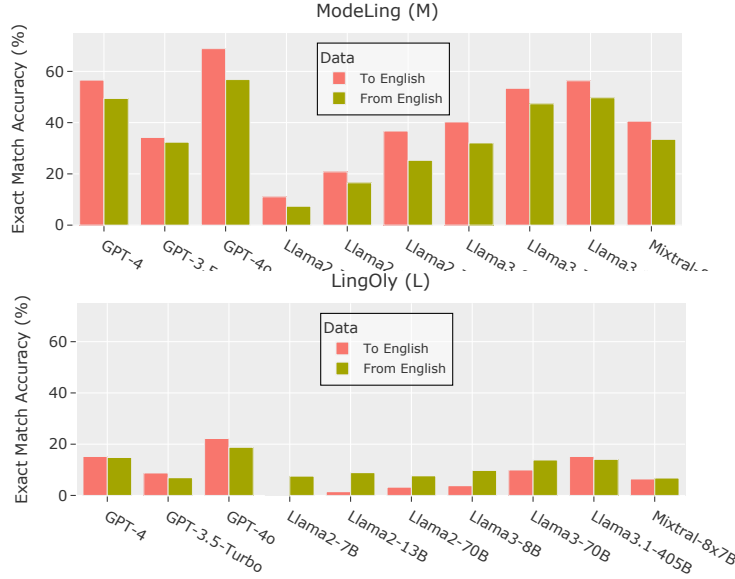


Figure 19: Models scores across translation directions.

Prompt	Previous Best	Score for DeepSeek-R1
NullPrompt	47.4	60.7
ModelLing Minimal	53.1	61.4
ModelLing Hand-Tuned	49.2	64.7
ModelLing Basic-CoT	48.7	64.9
ModelLing Full-CoT	49.0	60.5
LingOly Std	48.3	61.9

Table 8: Extended results for DeepSeek-R1 compared against the previous best scores across both datasets (O) for each prompt setting (see Table 7).

D Linguistic Features, Definitions, and Classification

See Tab. 9.

E Correlation Between Attributes and Model Scores

See Tab. 12.

F Annotation Guidelines & Inter-Annotator Agreement

Annotation guidelines are as follows:

- Solve the problem yourself if the solution is not detailed enough to list out all the features used in the problem
- Identify the linguistic features that **are being tested** or borrow them from the solution (if present), map them to the finer features list (add if it is not already there).
- If a new feature was added in the last step, map it into WALS categorisation and broad categorisation (based on general definition).
- Mark 0 for the feature in the data column if there are data points that can be removed from the problem without affecting the solvability for that feature, 1 otherwise

WALS Feature	WALS Category	Broad category	Definition	Example Problem	Example Problem Citation	Other Research
59 - Possessive Classification (<i>alien</i>)	Nominal Syntax	Semantics	the possessed object could be either close to the possessor (kinship, body part terms, etc.) or not	M.Abun	A Description of Abun: A West Papuan Language of Irian Jaya (5.3.1 and 5.3.2)	WALS - Poss. Class.
57 - Position of Poss. Markers (<i>poslinker</i>)	Nominal Syntax	Syntax	a possessive particle that between two nouns	M.Abun	A Description of Abun: A West Papuan Language of Irian Jaya (5.3.1 and 5.3.2)	Linkers and agreement WALS - Pos. of Poss. Markers
13 - Tone (<i>tone</i>)	Phonology	Phonology	pitch pattern grammar	M.Dogon	A Grammar of Dogon	WALS - Tone
101 - Expression of Pronominal Subjects (<i>pron</i>)	Simple Clauses	Syntax	phonological rules or Lexical representation changed by pronoun role	M.Dogon	A Grammar of Dogon (6.2.1.1)	WALS - Expt. of Pron. Subj.
33 - Nominal Plurality (<i>pl</i>)	Nominal Categories	Syntax	marker indicating plurality of noun	M.Rapa_Nui.6	A grammar of Rapa Nui - Paulus Kieviet 5.5	WALS - Nom. Pl.
102 - Verbal Person Marking (<i>transfix</i>)	Simple Clauses	Morphology	affix to mark transitivity (agent+patient) on verb	M.Ayutla_Mixe	A reference grammar of Ayutla Mixe (Tukyo'm ayuujk) (6.3.3.1 and 8.2.4)	WALS - Verb. Pers. Marking
102 - Verbal Person Marking (<i>persfix</i>)	Simple Clauses	Morphology	affixes to mark person (only agent) on verb	M.Ayutla_Mixe	A reference grammar of Ayutla Mixe (Tukyo'm ayuujk) 8.2.1	WALS - Verb. Pers. Marking
81-82 - Order of SOV, SV (<i>sv/cso/vs/vos...</i>)	Word Order	Syntax	order of subject, object, and verb in a sentence. + orders for clauses without objects	M.Ayutla_Mixe	A reference grammar of Ayutla Mixe (Tukyo'm ayuujk) 9.3.1.2	WALS - Order of SOV WALS - Order of SV
87 - Order of Adj., Noun (<i>nounadj/adjnoun</i>)	Word Order	Syntax	order of noun and adjective in a noun phrase	M.Bangime.6	Bangi me, a language of unknown affiliation 5.2	WALS - Order of N-A
NA - Diminutives (<i>dim</i>)	Nominal Categories	Semantics	marker to express the little modifier to a noun / verb	M.Kutenai	Kutenai lexicon	Intro: Dim. across languages, theories ... (WALS doesn't classify it)
112 - Negative Morphemes (<i>neg</i>)	Simple Clauses	Semantics	particle to express negation/absence of a noun	M.Kutenai	Kutenai lexicon	WALS - Neg. Morph.
NA - Other Semantic Affixes (<i>semfix</i>)	Nominal Categories	Morphology	affixes that add to the meaning to lexical categories	M.Guugu_Yimith	Lang. & Cognition: The Cog. Consequences of Spatial Description in Guugu Yimithir pg. 4	Prefixed of Spatiality in English: A Study in Cognitive Linguistics
55 - Numeral Classifiers (<i>cls</i>)	Nominal Categories	Semantics	classifier marker for nouns	M.Ngadha.2	NGADHA TEXT TRADITION - Stephanus Djawana 2.2.3	Nominal Classification / WALS - Num. Class.
131 - Numeral Bases (<i>num</i>)	Lexicon	Semantics	involves counting, numerals, and how positive whole number expressions are formed	M.Ngadha.2	NGADHA TEXT TRADITION - Stephanus Djawana 3.5.1.3	Numeral Systems / WALS - Num. Bases
NA - Compounds (<i>comp</i>)	Morphology	Semantics	compositionality: meanings of parts in an expression combine to convey something bigger	M.Kalam	On the origins of serial verb constructions in Kalam	Attributive Compounds (WALS doesn't classify this)
125 - Purpose Clauses (<i>svc</i>)	Complex Sentences	Syntax	serial verb construction: verbs combine in a single clause without any morphosyntactic marking of linking or subordination	M.Kalam	On the origins of serial verb constructions in Kalam	Serial Verb Constructions WALS - Purp. Clause.
24 - Possessive Noun Phrases (<i>nounposs</i>)	Nominal Syntax	Syntax	order of noun and possessive marker in a noun phrase	M.Seri.4	Seri Grammar	WALS - Poss. NP
85 - Order of Adposition and Noun Phrase (<i>nounprep...</i>)	Word Order	Syntax	order of noun and preposition in a prepositional phrase	M.Seri.2	Seri Grammar	WALS - Order of Adp. & NP
60 - Adjectival Clauses (<i>nounadjc</i>)	Nominal Syntax	Syntax	order of noun and adjectival clause in a noun phrase being noun then adjectival clause	M.Seri.2	Seri Grammar 2.4	WALS - Gen., Adj. and Rel. Cls.
24 - Possessive Noun Phrases (<i>possfix</i>)	Nominal Syntax	Morphology	possessives marked as affix	M.Totonac	The phonology and morphology of Filomeno Mata Totonac 3.3.2	WALS - Poss. Noun Phr.
32 - Gender (<i>genfix</i>)	Nominal Categories	Morphology	gender marking affixes	L.Beja	UKLO - Beja	WALS - Gender

117 - Predicative Possession (<i>poss</i>)	Simple Clauses	Syntax	possessive particles	L.Coptic	UKLO - Coptic	WALS - Pred. Poss.
69 - Position of Tense-Aspect Affixes (<i>tensefix</i>)	Verbal Categories	Morphology	tense marking affixes	L.K'iche'	UKLO - K'iche'	WALS - Pos. of Tense-Asp. Aff.
47 - Intensifiers & Reflexive Pronouns (<i>intens</i>)	Nominal Categories	Syntax	particle for intensifiers	L.K'iche'	UKLO - K'iche'	"That's proper cool": The emerging intensifier proper in British English / WALS - Int. & Refl. Pro.
NA - Elision (<i>eli</i>)	Morphology	Phonology	when a sound is elided during a fusion process, morphophonologically	L.Kabyle	UKLO - Kabyle	Schwa Elision in Fast Speech: Segmental Deletion or Gestural Overlap? WALS <i>doesn't classify this</i>
NA - Kinship order markers (<i>moiety</i>)	Nominal Categories	Semantics	the concept of moiety encoded in affixes	L.Lardil	UKLO - Lardil	7. Semantics of Australian Languages WALS <i>doesn't classify this</i>
34 - Nominal Plurality (<i>numfix</i>)	Nominal Categories	Morphology	number of subject/object marking affixes	L.Mayangna	UKLO - Mayangna	WALS - Occ. of Nom. Pl.
49 - Number of Cases (<i>case</i>)	Nominal Categories	Morphology	case marking	L.Nhanda	UKLO - Nhanda	WALS - No. of Cases
37 - Definite Articles (<i>def</i>)	Nominal Categories	Morphology	definitiveness marking affixes	L.Nhanda	UKLO - Nhanda	WALS - Def. Art. (the affixes bit)
112 - Negative Morphemes (<i>negfix</i>)	Simple Clauses	Morphology	negation marking affixes	L.Tadaksahak	UKLO - Tadaksahak	WALS - Neg. Morph.
107 - Passive Constructions (<i>voice</i>)	Simple Clauses	Syntax	active/passive voice having a morpho-syntactic impact	L.Tadaksahak	UKLO - Tadaksahak	WALS - Pass. Con.
32 - Systems of Gender Assignment (<i>anim</i>)	Nominal Categories	Semantics	whether noun has animacy or not	L.Taos	UKLO - Taos	Cog. anim. and its rel. to ling. anim. / WALS - Sys. of Gen.
27 - Reduplication (<i>redup</i>)	Morphology	Morphology	grammatical partial/full reduplication of syllables	L.Tawala	UKLO - Tawala	WALS - Reduplication
NA - Other Syntax Rules (<i>miscsyn</i>)	Complex Sentences	Syntax	other syntactic rules like semantic placement of negation particles, or having an aspect marking particle at the end of each sentence	L.Tawala	UKLO - Tawala	
NA - Other Morphology Rules (<i>miscfix</i>)	Morphology	Morphology	other affixes like "normalizer"	L.Tseltal	UKLO - Tseltal	
NA - Other Phonology Rules (<i>miscphon</i>)	Phonology	Phonology	other phonological features like assimilation	L.Waanyi	UKLO - Waanyi	
75 - Epistemic Possibility (<i>modfix</i>)	Verbal Categories	Morphology	modality marking affixes	L.Zou	UKLO - Zou	WALS - Epi. Poss.
51 - Position of Case Affixes (<i>preffix</i>)	Nominal Categories	Morphology	prepositional affixes	L.Zou	UKLO - Zou	WALS - Pos. of Case Aff. (prep clitics bit)
116 - Polar Questions (<i>quesfix</i>)	Simple Clauses	Morphology	interrogation affixes	L.Zou	UKLO - Zou	WALS - Pol. Ques.
95 - Order of Obj. & Verb and Adp. & NP (<i>prepc</i>)	Word Order	Syntax	prepositional syntax with respect to the head clause	M.Ainu	(PDF) Ainu applicatives in typological perspective	WALS - Relation bw Order of Obj. & Verb and Adp. & NP
2 - Vowel Quality (<i>vowel</i>)	Phonology	Phonology	phonological rules based on vowel quality or class.	L.Ulwa	Essentials of Linguistics: Describing vowels	WALS - Vowel Qual.
10 - Vowel Nasalisation (<i>nasal</i>)	Phonology	Phonology	phonological rules based on nasal consonants or sounds.	L.Coptic	Nasals and Nasalization	WALS - Vowel Nasal.

Table 9: Linguistic Feature tags used, their respective categorisations, definitions, example problem in the dataset, citation for the feature's presence in the example problem language, and related research on it.

#	Feature	κ	#	Feature	κ	#	Feature	κ	#	Feature	κ	#	Feature	κ
1	<i>adjnoun</i>	1.00	11	<i>genfix</i>	0.49	21	<i>nounadjc</i>	1.00	31	<i>possfix</i>	1.00	41	<i>svc</i>	1.00
2	<i>alien</i>	1.00	12	<i>intens</i>	0.85	22	<i>nounposs</i>	1.00	32	<i>posslinker</i>	1.00	42	<i>svo</i>	1.00
3	<i>anim</i>	1.00	13	<i>miscfix</i>	1.00	23	<i>nounprep</i>	1.00	33	<i>prepc</i>	0.66	43	<i>tensefix</i>	0.84
4	<i>case</i>	1.00	14	<i>miscphon</i>	0.79	24	<i>num</i>	1.00	34	<i>preprefix</i>	1.00	44	<i>vso</i>	1.00
5	<i>cls</i>	0.90	15	<i>miscsyn</i>	1.00	25	<i>numfix</i>	0.94	35	<i>prepnoun</i>	1.00	45	<i>tone</i>	1.00
6	<i>comp</i>	1.00	16	<i>modfix</i>	1.00	26	<i>persfix</i>	0.93	36	<i>quesfix</i>	0.66	46	<i>transfix</i>	0.91
7	<i>def</i>	0.88	17	<i>moiety</i>	1.00	27	<i>vowel</i>	0.49	37	<i>redup</i>	0.73	47	<i>voice</i>	1.00
8	<i>nasal</i>	0.66	18	<i>neg</i>	1.00	28	<i>pl</i>	1.00	38	<i>semfix</i>	1.00	48	<i>vos</i>	1.00
9	<i>dim</i>	1.00	19	<i>negfix</i>	1.00	29	<i>pron</i>	0.88	39	<i>sov</i>	0.95	49	<i>vs</i>	1.00
10	<i>eli</i>	1.00	20	<i>nounadj</i>	1.00	30	<i>poss</i>	1.00	40	<i>sv</i>	1.00	50		0.00

Table 10: Pre-adjudication inter-annotator agreement (Cohen’s κ) for marking feature presence in a puzzle. Note that after adjudication, all feature labelings have a perfect agreement ($\kappa=1.0$).

Puzzle	Exact-Match Score (+Gain)							
	Llama2-7B	Llama2-13B	Llama2-70B	Llama3-8B	Llama3-70B	Llama3.1-405B	GPT-4o	GPT-4
<i>L.Ilokano</i>	0.0 (+0.0)	0.0 (+0.0)	0.167 (+0.0)	0.0 (-0.167)	0.333 (+0.333)	0.167 (+0.167)	0.333 (+0.167)	0.167 (+0.167)
<i>L.Karelian</i>	0.0 (-0.077)	0.231 (-0.077)	0.462 (+0.231)	0.231 (-0.077)	0.769 (+0.154)	0.769 (-0.154)	0.692 (-0.077)	0.846 (+0.077)
<i>L.Lardil</i>	0.0 (+0.0)	0.167 (-0.167)	0.167 (+0.0)	0.333 (+0.333)	0.333 (+0.167)	0.167 (+0.0)	0.167 (-0.167)	0.167 (+0.0)
<i>M.Ayutla Mixe</i>	0.0 (+0.0)	0.0 (+0.0)	0.0 (-0.25)	0.0 (+0.0)	0.0 (+0.0)	0.0 (-0.25)	0.0 (+0.0)	0.0 (+0.0)
<i>M.Bangime_3</i>	0.0 (+0.0)	0.0 (+0.0)	0.0 (+0.0)	0.0 (+0.0)	0.4 (+0.2)	0.4 (+0.2)	0.6 (+0.2)	0.6 (+0.2)
<i>M.Bangime_5</i>	0.0 (+0.0)	0.4 (+0.4)	0.4 (+0.2)	0.2 (+0.2)	0.6 (+0.2)	0.8 (+0.4)	0.6 (+0.2)	0.8 (+0.2)
<i>M.Guugu.Yimithir</i>	0.0 (-0.1)	0.1 (+0.0)	0.3 (+0.1)	0.1 (-0.2)	0.7 (+0.2)	0.7 (+0.5)	0.7 (+0.2)	0.6 (+0.2)
<i>M.Kutenai</i>	0.2 (+0.0)	0.2 (-0.4)	0.4 (+0.2)	0.4 (+0.0)	1.0 (+0.6)	0.8 (+0.0)	0.6 (+0.2)	0.6 (+0.2)
<i>M.Totonac</i>	0.167 (-0.167)	0.167 (+0.167)	0.167 (+0.167)	0.0 (+0.0)	0.333 (+0.167)	0.333 (+0.167)	0.333 (+0.167)	0.167 (-0.333)

Table 11: Puzzle-wise score gain on marking morpheme boundaries with whitespaces

- If the feature is something that English also has/uses, mark 1. If English has a similar feature to the same end, mark 1 there instead
- Use references linked in the last two columns for any confusion (here look at App. D)

See Tab. 10. for inter-annotator agreement scores.

G Impact of Morpheme-Separated Input

See Tab. 11.

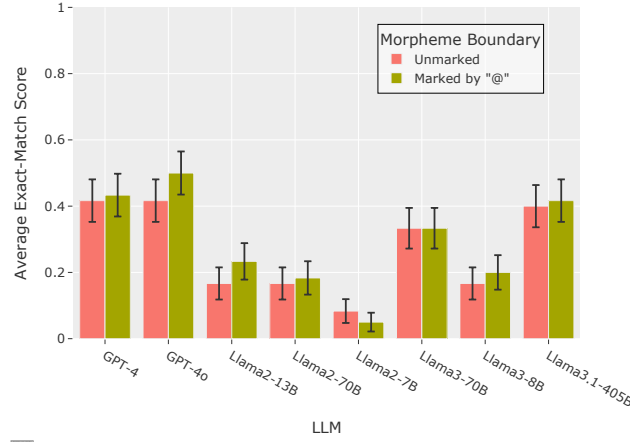


Figure 20: Performance change in models on making morpheme boundary explicit using "@"

		Llama2-7B	Llama2-13B	Llama2-70B	Mixtral-8x7B	Llama3-8B	Llama3-70B	Llama3.1-405B	GPT-3.5-Turbo	GPT-4o	GPT-4
Average Exact-Match Score		0.074	0.143	0.224	0.271	0.269	0.381	0.405	0.248	0.488	0.408
Attribute	Prompt Setting										
avg_eng	LingOlyStdPromptBuilder	0.031	0.053	0.388**	0.3*	0.491***	0.364**	0.276*	0.399**	0.258*	0.381**
	ModelingBasicCoTPromptBuilder	-0.005	0.2	0.387**	0.401**	0.427***	0.385**	0.257*	0.52**	0.293*	0.379**
	ModelingFullCoTPromptBuilder	0.021	0.365**	0.246	0.339**	0.431***	0.343**	0.135	0.341**	0.26*	0.304*
	ModelingHandTunedPromptBuilder	0.026	0.46*	0.427**	0.483**	0.395*	0.408**	0.25*	0.369**	0.293*	0.385**
	ModelingMinimalPromptBuilder	0.16	0.385**	0.37**	0.385**	0.316*	0.468**	0.342**	0.406**	0.197	0.365**
	NullPromptBuilder	-0.207	-0.182	0.224	0.418***	0.354**	0.42**	0.117	0.259*	0.253*	0.296*
	Overall	0.031	0.053	0.388**	0.3*	0.491***	0.364**	0.276*	0.399**	0.258*	0.381**
avg_eng_sim	LingOlyStdPromptBuilder	-0.01	-0.179	-0.283*	-0.206	-0.385**	-0.265*	-0.146	-0.216	-0.092	-0.224
	ModelingBasicCoTPromptBuilder	-0.055	-0.239	-0.342**	-0.331*	-0.361*	-0.301*	-0.133	-0.398*	-0.148	-0.283*
	ModelingFullCoTPromptBuilder	-0.194	-0.398**	-0.22	-0.24	-0.371**	-0.242	0.002	-0.098	-0.145	-0.203
	ModelingHandTunedPromptBuilder	-0.211	-0.405***	-0.307*	-0.368**	-0.25*	-0.386**	-0.084	-0.179	-0.15	-0.304*
	ModelingMinimalPromptBuilder	-0.17	-0.272*	-0.264*	-0.271*	-0.188	-0.318*	-0.218	-0.301*	-0.021	-0.248*
	NullPromptBuilder	-0.059	-0.059	-0.102	-0.34**	-0.348**	-0.248*	-0.006	-0.179	0.035	-0.099
	Overall	-0.01	-0.179	-0.283*	-0.206	-0.385**	-0.265*	-0.146	-0.216	-0.092	-0.224
count_exacts	LingOlyStdPromptBuilder	-0.144	-0.198	-0.448***	-0.466***	-0.467***	-0.55***	-0.592***	-0.543***	-0.572***	-0.547***
	ModelingBasicCoTPromptBuilder	-0.238	-0.287*	-0.454***	-0.477***	-0.46***	-0.539***	-0.615***	-0.495***	-0.603***	-0.533***
	ModelingFullCoTPromptBuilder	-0.208	-0.335**	-0.457***	-0.502***	-0.451***	-0.569***	-0.674***	-0.617***	-0.577***	-0.585***
	ModelingHandTunedPromptBuilder	-0.17	-0.38**	-0.457***	-0.533***	-0.496***	-0.512***	-0.643***	-0.623***	-0.619***	-0.511***
	ModelingMinimalPromptBuilder	-0.271*	-0.399**	-0.472**	-0.534**	-0.537**	-0.53**	-0.359**	-0.504**	-0.649**	-0.597**
	NullPromptBuilder	-0.064	-0.09	-0.414***	-0.481***	-0.279*	-0.544***	-0.346**	-0.376**	-0.668***	-0.612***
	Overall	-0.144	-0.198	-0.448***	-0.466***	-0.467***	-0.55***	-0.592***	-0.543***	-0.572***	-0.547***
ft_count	LingOlyStdPromptBuilder	-0.17	-0.263*	-0.497***	-0.544***	-0.543***	-0.609***	-0.648***	-0.548***	-0.621***	-0.567***
	ModelingBasicCoTPromptBuilder	-0.303*	-0.38**	-0.531***	-0.521***	-0.523***	-0.594***	-0.67***	-0.491***	-0.627***	-0.566***
	ModelingFullCoTPromptBuilder	-0.25*	-0.415***	-0.509***	-0.531***	-0.541***	-0.635***	-0.742***	-0.631***	-0.636***	-0.621***
	ModelingHandTunedPromptBuilder	-0.254*	-0.464***	-0.523***	-0.574***	-0.566***	-0.581***	-0.69***	-0.631***	-0.659***	-0.556***
	ModelingMinimalPromptBuilder	-0.284*	-0.448***	-0.517***	-0.585***	-0.627***	-0.57***	-0.389**	-0.545***	-0.709***	-0.631***
	NullPromptBuilder	-0.124	-0.148	-0.479**	-0.518*	-0.382*	-0.599**	-0.403*	-0.326*	-0.7*	-0.617**
	Overall	-0.17	-0.263*	-0.497***	-0.544***	-0.543***	-0.609***	-0.648***	-0.548***	-0.621***	-0.567***
s_Morphology	LingOlyStdPromptBuilder	-0.156	-0.261*	-0.482***	-0.558**	-0.554**	-0.604***	-0.652***	-0.514***	-0.595***	-0.507***
	ModelingBasicCoTPromptBuilder	-0.281*	-0.377**	-0.507***	-0.52**	-0.494**	-0.562***	-0.666***	-0.444***	-0.57***	-0.523***
	ModelingFullCoTPromptBuilder	-0.234	-0.421**	-0.507***	-0.535***	-0.544**	-0.609***	-0.729***	-0.618***	-0.618***	-0.585***
	ModelingHandTunedPromptBuilder	-0.296	-0.469**	-0.497***	-0.569**	-0.565**	-0.554**	-0.688**	-0.581**	-0.643**	-0.523**
	ModelingMinimalPromptBuilder	-0.262*	-0.426**	-0.528**	-0.585**	-0.614**	-0.565**	-0.405*	-0.564**	-0.707**	-0.605**
	NullPromptBuilder	-0.094	-0.106	-0.477**	-0.543**	-0.377**	-0.611**	-0.399**	-0.283*	-0.691**	-0.581**
	Overall	-0.219**	-0.348**	-0.496**	-0.546**	-0.522**	-0.583**	-0.572**	-0.483**	-0.637**	-0.553**
s_Phonology	LingOlyStdPromptBuilder	0.001	-0.05	-0.16	-0.25*	-0.218	-0.227	-0.243	-0.235	-0.341**	-0.273*
	ModelingBasicCoTPromptBuilder	-0.044	-0.017	-0.225	-0.233	-0.216	-0.221	-0.228	-0.2	-0.289*	-0.227
	ModelingFullCoTPromptBuilder	-0.039	-0.093	-0.226	-0.191	-0.169	-0.253*	-0.215	-0.301*	-0.275*	-0.326*
	ModelingHandTunedPromptBuilder	-0.035	-0.137	-0.168	-0.218	-0.288*	-0.179	-0.257*	-0.272*	-0.3*	-0.259*
	ModelingMinimalPromptBuilder	-0.172	-0.175	-0.203	-0.18	-0.307*	-0.215	-0.207	-0.2	-0.341**	-0.281*
	NullPromptBuilder	-0.063	-0.012	-0.178	-0.15	-0.156	-0.251*	-0.255*	-0.068	-0.355**	-0.264*
	Overall	0.001	-0.05	-0.16	-0.25*	-0.218	-0.227	-0.243	-0.235	-0.341**	-0.273*
s_Semantics	LingOlyStdPromptBuilder	-0.084	0.028	-0.31*	-0.215	-0.25*	-0.329**	-0.291*	-0.354**	-0.331**	-0.347**
	ModelingBasicCoTPromptBuilder	-0.02	-0.136	-0.199	-0.261*	-0.258*	-0.289*	-0.296*	-0.333*	-0.349**	-0.297*
	ModelingFullCoTPromptBuilder	0.047	-0.101	-0.206	-0.314*	-0.229	-0.302*	-0.311*	-0.369**	-0.304*	-0.252*
	ModelingHandTunedPromptBuilder	0.071	-0.181	-0.282*	-0.32*	-0.224	-0.239	-0.322**	-0.367**	-0.342**	-0.277*
	ModelingMinimalPromptBuilder	-0.084	-0.202	-0.271*	-0.293*	-0.212	-0.302*	-0.115	-0.258*	-0.341**	-0.336**
	NullPromptBuilder	0.142	0.078	-0.227	-0.298*	-0.157	-0.305*	-0.067	-0.249*	-0.406**	-0.434**
	Overall	-0.084	0.028	-0.31*	-0.215	-0.25*	-0.329**	-0.291*	-0.354**	-0.331**	-0.347**
s_Syntax	LingOlyStdPromptBuilder	-0.083	-0.183	-0.073	-0.079	-0.074	-0.089	-0.118	-0.083	-0.087	-0.134
	ModelingBasicCoTPromptBuilder	-0.215	-0.157	-0.169	-0.067	-0.134	-0.164	-0.151	-0.1	-0.161	-0.157
	ModelingFullCoTPromptBuilder	-0.222	-0.159	-0.105	-0.044	-0.122	-0.154	-0.211	-0.057	-0.13	-0.174
	ModelingHandTunedPromptBuilder	-0.129	-0.118	-0.129	-0.075	-0.103	-0.197	-0.129	-0.144	-0.104	-0.132
	ModelingMinimalPromptBuilder	-0.101	-0.125	-0.047	-0.107	-0.169	-0.086	-0.061	-0.061	-0.095	-0.117
	NullPromptBuilder	-0.222	-0.236	-0.09	-0.023	-0.088	-0.056	-0.128	-0.096	-0.046	-0.062
	Overall	-0.083	-0.183	-0.073	-0.079	-0.074	-0.089	-0.118	-0.083	-0.087	-0.134
w_Complex_Sentences	LingOlyStdPromptBuilder	-0.081	-0.101	-0.14	-0.162	-0.154	-0.198	-0.198	-0.166	-0.215	-0.18
	ModelingBasicCoTPromptBuilder	-0.098	-0.116	-0.151	-0.157	-0.156	-0.199	-0.214	-0.143	-0.181	-0.192
	ModelingFullCoTPromptBuilder	-0.098	-0.13	-0.164	-0.164	-0.159	-0.192	-0.23	-0.188	-0.177	-0.205
	ModelingHandTunedPromptBuilder	-0.108	-0.138	-0.16	-0.165	-0.169	-0.191	-0.202	-0.183	-0.16	-0.185
	ModelingMinimalPromptBuilder	-0.105	-0.125	-0.153	-0.183	-0.178	-0.153	-0.155	-0.157	-0.175	-0.196
	NullPromptBuilder	-0.082	-0.088	-0.149	-0.143	-0.13	-0.17	-0.13	-0.085	-0.229	-0.204
	Overall	-0.081	-0.101	-0.14	-0.162	-0.154	-0.198	-0.198	-0.166	-0.215	-0.18

<i>w_Lexicon</i>	<i>LingOlyStdPromptBuilder</i>	-0.1	-0.125	-0.172	-0.182	-0.172	-0.203	-0.147	-0.153	-0.168	-0.168
	<i>ModelingBasicCoTPromptBuilder</i>	-0.121	-0.118	-0.187	-0.194	-0.174	-0.164	-0.157	-0.158	-0.191	-0.184
	<i>ModelingFullCoTPromptBuilder</i>	-0.085	-0.139	-0.182	-0.18	-0.197	-0.182	-0.182	-0.184	-0.154	-0.146
	<i>ModelingHandTunedPromptBuilder</i>	-0.093	-0.149	-0.177	-0.187	-0.179	-0.151	-0.158	-0.209	-0.171	-0.19
	<i>ModelingMinimalPromptBuilder</i>	-0.082	-0.127	-0.189	-0.18	-0.141	-0.144	-0.135	-0.143	-0.24	-0.163
	<i>NullPromptBuilder</i>	-0.064	-0.076	-0.184	-0.158	-0.137	-0.128	-0.024	-0.053	-0.153	-0.237
<i>Overall</i>		-0.1	-0.125	-0.172	-0.182	-0.172	-0.203	-0.147	-0.153	-0.168	-0.168
<i>w_Morphology</i>	<i>LingOlyStdPromptBuilder</i>	-0.035	-0.133	-0.245	-0.308*	-0.293*	-0.409***	-0.395**	-0.312*	-0.401**	-0.309*
	<i>ModelingBasicCoTPromptBuilder</i>	-0.024	-0.211	-0.323**	-0.277*	-0.285*	-0.4*	-0.434***	-0.348**	-0.428***	-0.339**
	<i>ModelingFullCoTPromptBuilder</i>	-0.109	-0.219	-0.324**	-0.302*	-0.286*	-0.396**	-0.395**	-0.415***	-0.437***	-0.383**
	<i>ModelingHandTunedPromptBuilder</i>	-0.136	-0.253*	-0.302*	-0.35**	-0.306*	-0.379**	-0.396**	-0.418***	-0.408***	-0.351**
	<i>ModelingMinimalPromptBuilder</i>	-0.171	-0.232	-0.272*	-0.364**	-0.369**	-0.365**	-0.17	-0.356**	-0.424**	-0.403***
	<i>NullPromptBuilder</i>	0.038	0.018	-0.299*	-0.343**	-0.246*	-0.359**	-0.285*	-0.235	-0.401**	-0.419**
<i>Overall</i>		-0.035	-0.133	-0.245	-0.308*	-0.293*	-0.409***	-0.395**	-0.312*	-0.401**	-0.309*
<i>w_Nominal_Categories</i>	<i>LingOlyStdPromptBuilder</i>	-0.13	-0.033	-0.399**	-0.377**	-0.374**	-0.397**	-0.426***	-0.422***	-0.464***	-0.413***
	<i>ModelingBasicCoTPromptBuilder</i>	-0.16	-0.202	-0.282*	-0.382**	-0.329**	-0.343**	-0.433***	-0.333**	-0.45***	-0.352**
	<i>ModelingFullCoTPromptBuilder</i>	-0.015	-0.206	-0.285*	-0.423***	-0.334**	-0.382**	-0.472***	-0.469***	-0.378**	-0.38**
	<i>ModelingHandTunedPromptBuilder</i>	-0.038	-0.278*	-0.337**	-0.423***	-0.364**	-0.31*	-0.472***	-0.417***	-0.479***	-0.357**
	<i>ModelingMinimalPromptBuilder</i>	-0.053	-0.282*	-0.352**	-0.414***	-0.352**	-0.399**	-0.247*	-0.377**	-0.522**	-0.448***
	<i>NullPromptBuilder</i>	0.062	0.027	-0.272*	-0.405***	-0.215	-0.416***	-0.138	-0.239	-0.566***	-0.478***
<i>Overall</i>		-0.13	-0.033	-0.399**	-0.377**	-0.374**	-0.397**	-0.426***	-0.422***	-0.464***	-0.413***
<i>w_Nominal_Syntax</i>	<i>LingOlyStdPromptBuilder</i>	0.225	0.034	-0.033	-0.057	-0.087	-0.111	-0.131	-0.182	-0.122	-0.152
	<i>ModelingBasicCoTPromptBuilder</i>	0.033	0.019	-0.102	0.034	-0.094	-0.144	-0.201	-0.18	-0.228	-0.164
	<i>ModelingFullCoTPromptBuilder</i>	0	0.002	-0.033	-0.024	-0.086	-0.172	-0.265*	-0.221	-0.184	-0.197
	<i>ModelingHandTunedPromptBuilder</i>	0.175	-0.013	-0.116	-0.069	-0.065	-0.153	-0.213	-0.276*	-0.151	-0.119
	<i>ModelingMinimalPromptBuilder</i>	-0.056	-0.055	-0.019	-0.109	-0.17	-0.129	0.004	-0.146	-0.154	-0.146
	<i>NullPromptBuilder</i>	0.149	0.115	-0.142	-0.029	0.089	-0.167	-0.225	-0.157	-0.187	-0.171
<i>Overall</i>		0.225	0.034	-0.033	-0.057	-0.087	-0.111	-0.131	-0.182	-0.122	-0.152
<i>w_Phonology</i>	<i>LingOlyStdPromptBuilder</i>	0.021	-0.027	-0.132	-0.221	-0.19	-0.188	-0.204	-0.243	-0.327**	-0.24
	<i>ModelingBasicCoTPromptBuilder</i>	-0.022	0.011	-0.197	-0.203	-0.186	-0.181	-0.24	-0.173	-0.273*	-0.218
	<i>ModelingFullCoTPromptBuilder</i>	-0.016	-0.065	-0.195	-0.158	-0.137	-0.216	-0.226	-0.268*	-0.26*	-0.289*
	<i>ModelingHandTunedPromptBuilder</i>	-0.009	-0.108	-0.135	-0.186	-0.259*	-0.139	-0.241	-0.239	-0.251*	-0.224
	<i>ModelingMinimalPromptBuilder</i>	-0.153	-0.152	-0.173	-0.177	-0.276*	-0.178	-0.269*	-0.17	-0.322**	-0.244
	<i>NullPromptBuilder</i>	-0.046	0.009	-0.149	-0.121	-0.13	-0.22	-0.233	-0.05	-0.344**	-0.255*
<i>Overall</i>		0.021	-0.027	-0.132	-0.221	-0.19	-0.188	-0.204	-0.243	-0.327**	-0.24
<i>w_Simple_Clauses</i>	<i>LingOlyStdPromptBuilder</i>	-0.207	-0.33**	-0.467***	-0.518***	-0.553***	-0.566***	-0.625***	-0.443***	-0.47***	-0.436***
	<i>ModelingBasicCoTPromptBuilder</i>	-0.316*	-0.417***	-0.5***	-0.496***	-0.486***	-0.551***	-0.57***	-0.389**	-0.412***	-0.478***
	<i>ModelingFullCoTPromptBuilder</i>	-0.277*	-0.457***	-0.505***	-0.485***	-0.552***	-0.583***	-0.659***	-0.529***	-0.572***	-0.508***
	<i>ModelingHandTunedPromptBuilder</i>	-0.397**	-0.509***	-0.502***	-0.517***	-0.56***	-0.574***	-0.62***	-0.496***	-0.566***	-0.469***
	<i>ModelingMinimalPromptBuilder</i>	-0.362**	-0.447***	-0.559***	-0.529***	-0.618***	-0.528***	-0.356**	-0.523***	-0.583***	-0.517***
	<i>NullPromptBuilder</i>	-0.195	-0.195	-0.486***	-0.486***	-0.408***	-0.578***	-0.424***	-0.244	-0.559***	-0.447***
<i>Overall</i>		-0.207	-0.33**	-0.467***	-0.518***	-0.553***	-0.566***	-0.625***	-0.443***	-0.47***	-0.436***
<i>w_Verbal_Categories</i>	<i>LingOlyStdPromptBuilder</i>	-0.194	-0.243	-0.282*	-0.333**	-0.333**	-0.39**	-0.422***	-0.308*	-0.392**	-0.354**
	<i>ModelingBasicCoTPromptBuilder</i>	-0.234	-0.278*	-0.343**	-0.346**	-0.322**	-0.372**	-0.421***	-0.247*	-0.356**	-0.333**
	<i>ModelingFullCoTPromptBuilder</i>	-0.235	-0.295*	-0.35**	-0.328**	-0.35**	-0.38**	-0.508***	-0.357**	-0.404**	-0.359**
	<i>ModelingHandTunedPromptBuilder</i>	-0.26*	-0.3*	-0.326**	-0.34**	-0.355**	-0.362**	-0.417***	-0.347**	-0.396**	-0.299*
	<i>ModelingMinimalPromptBuilder</i>	-0.252*	-0.279*	-0.355**	-0.341**	-0.375**	-0.345**	-0.328**	-0.292*	-0.43***	-0.373**
	<i>NullPromptBuilder</i>	-0.15	-0.15	-0.288*	-0.343**	-0.283*	-0.36**	-0.244	-0.203	-0.443***	-0.353**
<i>Overall</i>		-0.194	-0.243	-0.282*	-0.333**	-0.333**	-0.39**	-0.422***	-0.308*	-0.392**	-0.354**
<i>w_Word_Order</i>	<i>LingOlyStdPromptBuilder</i>	-0.254*	-0.271*	-0.034	-0.019	0.02	0.003	0	0.101	0.052	-0.005
	<i>ModelingBasicCoTPromptBuilder</i>	-0.313*	-0.221	-0.12	-0.082	-0.081	-0.058	0.002	0.032	0.021	-0.045
	<i>ModelingFullCoTPromptBuilder</i>	-0.333**	-0.207	-0.092	-0.011	-0.068	-0.046	0.002	0.172	0.009	-0.049
	<i>ModelingHandTunedPromptBuilder</i>	-0.319*	-0.116	-0.027	-0.016	-0.037	-0.086	0.034	0.079	0.028	-0.067
	<i>ModelingMinimalPromptBuilder</i>	-0.054	-0.085	0.001	-0.005	-0.04	0.034	-0.034	0.06	0.041	-0.003
	<i>NullPromptBuilder</i>	-0.433***	-0.432***	0.02	0.031	-0.159	0.087	0.037	0.047	0.166	0.095
<i>Overall</i>		-0.254*	-0.271*	-0.034	-0.019	0.02	0.003	0	0.101	0.052	-0.005

Table 12: Pearson correlation values between exact-match scores and feature values for all features against all model and prompt-setting combinations. Correlation values marked with “*”, “**”, and “***” have a corresponding p -value less than 0.05, 0.01, and 0.001, respectively.