MLissard: Multilingual Long and Simple Sequential Reasoning Benchmarks

Mirelle Bueno, Roberto Lotufo, Rodrigo Nogueira

School of Electrical and Computing Engineering,
State University of Campinas (UNICAMP),
m174909@dac.unicamp.br, {lotufo,rfn}@unicamp.br

Abstract

Language models are now capable of solving tasks that require dealing with long sequences consisting of hundreds of thousands of tokens. However, they often fail on tasks that require repetitive use of simple rules, even on sequences that are much shorter than those seen during training. For example, state-of-the-art LLMs can find common items in two lists with up to 20 items but fail when lists have 80 items. In this paper, we introduce MLissard, a multilingual benchmark designed to evaluate models' abilities to process and generate texts of varied lengths and offers a mechanism for controlling sequence complexity.

Our evaluation of open-source and proprietary models show a consistent decline in performance across all models and languages as the complexity of the sequence increases. Surprisingly, the use of in-context examples in languages other than English helps increase extrapolation performance significantly. The datasets and code are available at https://github.com/unicamp-dl/Lissard

1 Introduction

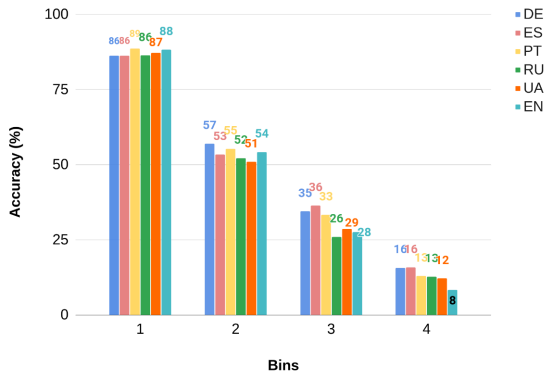

Figure 1: Performance of GPT-4 on the MLissard benchmark. See Table 2 for the definition of the bins.

The efficacy of language models, particularly in reasoning tasks, is significantly impacted by longer text lengths than those seen in training (Li et al., 2023b; Liu et al., 2024; Lake and Baroni, 2018). This phenomenon, referred to as "Length Generalization" or "Length Extrapolation" in the literature (Press et al., 2022; Zhao et al., 2023), is also common in models based on the Transformer architecture (Liška et al., 2018; Lewkowycz et al., 2022; Delétang et al., 2023; Zhou et al., 2023b). Notably, even Large Language Models (LLMs), known for their strong performance in a wide range of tasks and domains, are not immune to this problem (Anil et al., 2022; Chen et al., 2023).

Recent research tried to address this challenge by modifications to the positional embeddings (Press et al., 2022; Chi et al., 2022, 2023; Li et al., 2023b; Ke et al., 2021) or by using prompting strategies such as scratchpad (Nye et al., 2021) and chain-of-thought reasoning (Wei et al., 2022). Nevertheless, there remains a lack of datasets specifically designed for the systematic evaluation of the problem.

While benchmarks such as ZeroSCROLLS (Shaham et al., 2023) and InfiniteBench(Zhang et al., 2024) were designed to evaluate models in natural language tasks that involve long sequences, its effectiveness in monitoring model performance degradation within the context of length generalization may be limited by lack of explicit control of task complexity with respect to sequence length. For example, when using natural language texts there is no guarantee that answering a question about a longer text is harder than responding to one about a shorter text. This limitation highlights the need for benchmarks that can explicitly manipulate and test the impact of sequence length on model performance. In benchmarks pertaining to dialogues (Li et al., 2023a) and multi-document question answering (Liu et al., 2024), techniques like retrieval-augmented generation (RAG) are preva-

Proceedings of the 2nd GenBench Workshop on Generalisation (Benchmarking) in NLP, pages 86–95

lent, and therefore explicitly isolating the length extrapolation issue poses a challenge.

To address these aforementioned problems, we present MLissard, a multilingual benchmark that offer support for 6 languages (English, German, Portuguese, Russian, Spanish and Ukrainian) designed to evaluate the ability of models on tasks that require the use of repetitive simple rules, whose difficulty increases with respect to the sequence length. By incorporating varying degrees of difficulty within the same tasks, MLissard facilitates the identification of a models' breaking points. Given the syntactic nature of the datasets, researchers have the capability to generate new examples and increase the task difficulty, thus making it more challenging for newer and more capable models to be evaluated effectively. This flexibility also mitigates the contamination problem – where models may inadvertently be exposed to test datasets during their training (Ahuja et al., 2023; Li and Flanigan, 2024) – since synthetic datasets can be generated as needed, a advantage over traditional, manually curated datasets. At the time of this research, this is the first multilingual dataset designed to evaluate the quality of models in extrapolation via length.

Our analysis, which includes evaluations on proprietary models such as GPT-4 (OpenAI, 2023), as well as open-source ones like Llama-3 (Dubey et al., 2024), reveals a common trend among them. As illustrated in Figure 1, our findings underscore that irrespective of their architectures and parameter counts, all examined models demonstrate a performance degradation with increasing length, controlled by the number of key entities (see their definition in Table 2), required to solve the tasks. This indicates a common point of failure in generalization for LLMs, even for sequence lengths that are considerably shorter in terms of tokens than those seen during their pretraining or fine-tuning phases.

Our findings further demonstrated that the effect of extrapolation is not isolated; variables such as language and model size significantly influence the outcomes. For instance, despite English being a high-resource language, its performance was only average and was surpassed by other languages such as German. Moreover, ablation tests revealed improvements in extrapolation performance when in-context examples comprised a mixture of languages. This underscores the influence of language selection on the extrapolation capabilities of language models.

2 Related Work

The challenge of length extrapolation in the domain of natural language processing has been a persistent and long-standing issue. An array of studies has demonstrated that neural architectures encounter difficulties when confronted with sequences of longer than those they encountered during their training (Lake and Baroni, 2018; Liška et al., 2018; Keysers et al., 2019; Dubois et al., 2020; Nogueira et al., 2021; Welleck et al., 2022; Lewkowycz et al., 2022; Delétang et al., 2023; Zhou et al., 2023b). Despite efforts to expand the context window in LLMs, this issue persists, particularly when tackling tasks involving complex reasoning (Anil et al., 2022).

Recent endeavors have been undertaken to enhance the general performance of LLMs by employing prompt engineering techniques and by developing novel decoding methods aimed at expanding their capacity to extrapolate effectively over lengthy sequences of tokens. For instance, Nye et al. introduced the concept of a "scratchpad" that enables the model to generate draft responses in natural language before producing the final output. To assess the performance of this method, a range of tasks were employed, including math and coding tasks. Moreover, studies by Wei et al. and Zhou et al. demonstrated improvements by configuring the model to generate explanations for problem-solving and breaking down tasks into multiple interactive steps. These enhancements were particularly noticeable in tasks requiring the ability to extrapolate, such as SCAN (Lake and Baroni, 2018) (compositional generalization), and mathematical reasoning. Additionally, Bueno et al. showed that utilizing markups tokens as position representations help the model to generalize to longer sequences in tasks related to mathematical addition and compositional generalization. Han et al. devised a decoding method to improve generalization over extended sequences.

In addition to techniques for customizing prompts, recent research has explored modifying the position encoding function of the original transformer architecture to enhance its extrapolation capabilities (Press et al., 2022; Chi et al., 2022, 2023; Li et al., 2023b; Qin et al., 2023; Chen et al., 2023). For instance, Kazemnejad et al. conducted an evaluation of commonly used positional encoding

methods, finding that omitting positional encoding altogether yielded superior results in downstream tasks.

The studies cited above illustrate multiple methods designed to address the challenge of extrapolation. Nevertheless, there is a notable gap in research concerning the development of diverse and standardized datasets specifically for assessing the generation and synthesis of extended text sequences by neural models. This gap is particularly notable given that many of the traditional datasets may already have been employed in the training of large language models.

3 Datasets Description

Our benchmark incorporates a combination of existing tasks, such as those from BIG-bench (bench authors, 2023), as well as newly developed ones. The criteria for selecting tasks were based on their ease of solution, the ability to expand new examples of varying lengths via scripting, and their effectiveness in exercising reasoning and memorization.

We intentionally excluded classical datasets (e.g., SCAN) from the analysis since their test sets are publicly available and many solutions have been extensively detailed in scientific literature, potentially making them familiar to large language models (LLMs).

In addition to English (EN), the language set includes German (DE), Spanish (ES), Portuguese (PT), Russian (RU), and Ukrainian (UA). We achieved this expansion by integrating automatic translation systems and using Python scripts to generate synthetic data.

The following sections describe the idea of key entities, tasks, and how evaluation was performed.

3.1 Key entities

The notion of key entities functions as an extrapolation factor within the context of a target task. For instance, in a task that seeks to identify common items between two lists, this extrapolation factor is defined by the number of items the model requires to analyze. Utilizing this factor allows for the augmentation of task complexity without modifying its properties. As a result, within specified ranges (bins), we can identify the model's breakpoints.

The choice of bins for each task was designed to reflect different difficulty levels: short, intermediate, long, and super long, for example, Bin 1 consists of sequences of shorter length, while Bin 4 comprises sequences of longer length. Table 2 describes the key entities and the respective lengths in each bin. The values defining the intervals of each bin vary for each task and were empirically determined, inspired by BIG-bench tasks.

3.2 Tasks

In total, four tasks were developed, and Table 1 provides a summary of each one with input and output examples. Due to the high costs of paid APIs, we restricted our tests to 300 examples per task and language. To ensure balanced evaluations across different length partitions, we randomly selected 75 examples for each bin.

3.2.1 Object Counting

The main goal of this task is to assess the proficiency in object counting within sequences, as shown in Table 1. The input to the model is a sequence comprising a list of objects paired with their respective quantities and the expected output is a string with the total count of objects. Diverging from the original BIG-bench task that exclusively encompasses the enumeration of objects from predetermined categories like fruits, vegetables, or musical instruments, our method comprises object counting across different categories.

Automatic translation systems were used to generate the multilingual set, in this case, Google Translate. After this phase, a translation subset was selected for human analysis of the general quality of the translation.

3.2.2 List Intersection

The objective of this task is to find common items in two lists. Items within the lists are composed of words from a designated target language, with both the words and their frequencies sourced from the FrequencyWords[1] repository. For each specific language, stop words and special characters were eliminated. Following this preprocessing phase, a random sampling of words was conducted.

The lists have equal sizes, but the number of overlapping items varies. The target output is the words in common, sorted alphabetically. If there are no items in common, "None" must be returned.

3.2.3 Last Letter Concatenation

The Last Letter Concatenation task, as formulated in the Chain-of-Thought work (Wei et al., 2022), involves concatenating the last letter of each word

[1] https://github.com/hermitdave/FrequencyWords/

Task	Input Example	Output
Last Letter Concatenation	Abil Gaby	l y
Repeat Copy Logic	Repeat 2 times school	school school
Object Counting	I have a chair, and an apple.	2
List Intersection	A: abil,matt / B: matt, gaby	matt

Table 1: Task Summary in the MLissard Benchmark.

Task	Key Entity	Bin 1	Bin 2	Bin 3	Bin 4
LLC	Names	1-8	8-15	15-22	22-30
RCL	Total Repetitions	1-9	9-17	17-25	25-33
OC	Objects	1-7	7-12	12-17	17-23
LI	Items: lists A and B	1-46	46-91	91-136	136-181

Table 2: Key task entities: Last Letter Concatenation (LLC), Repeat Copy Logic (RCL), Object Counting (OC), and List Intersection (LI) and their respective ranges in each bin in Figure 1.

within an input sequence comprised of random names. Table 1 provides an illustrative instance of the dataset, where the input sequence comprises randomly selected names obtained through the target language Name Census[2].

In constructing our dataset, we applied a comparable methodology; however, we sampled the most common names from each target language and expanded the sample length to encompass sequences with an increase of up to thirty names.

[2]Portuguese (PT) - https://censo2010.ibge.gov.br/nomes/#/ranking
Spanish (ES) - https://www.epdata.es/datos/nombres-apellidos-mas-frecuentes-espana-ine/373
English (EN) - https://www.ssa.gov/cgi-bin/popularnames.cgi
German (DE) - http://www.firstnamesgermany.com/
Ukrainian (UA) - https://census.name/ukrainian-name-database/
Russsian (RU) - https://census.name/russian-name-database/

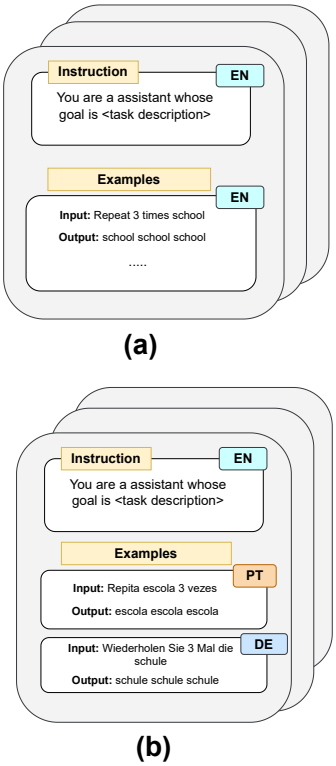

Figure 2: Template for evaluation. Being (a) Instruction and examples of tasks in the target language; (b) Instruction in the target language and multilingual examples.

3.2.4 Repeat Copy Logic

The task proposed by the BIG-bench evaluates language models' ability to comprehend and execute instructions involving repetitions, text-to-copy, basic logic, and conditionals, focusing on their extrapolation capabilities.

Our methodology for creating the dataset includes: i) Collecting responses to all input sequences from the BIG-bench repository[3]; ii) Filtering responses to retain only those correctly an-

[3]https://github.com/google/BIG-bench/tree/main/bigbench/benchmark_tasks/repeat_copy_logic

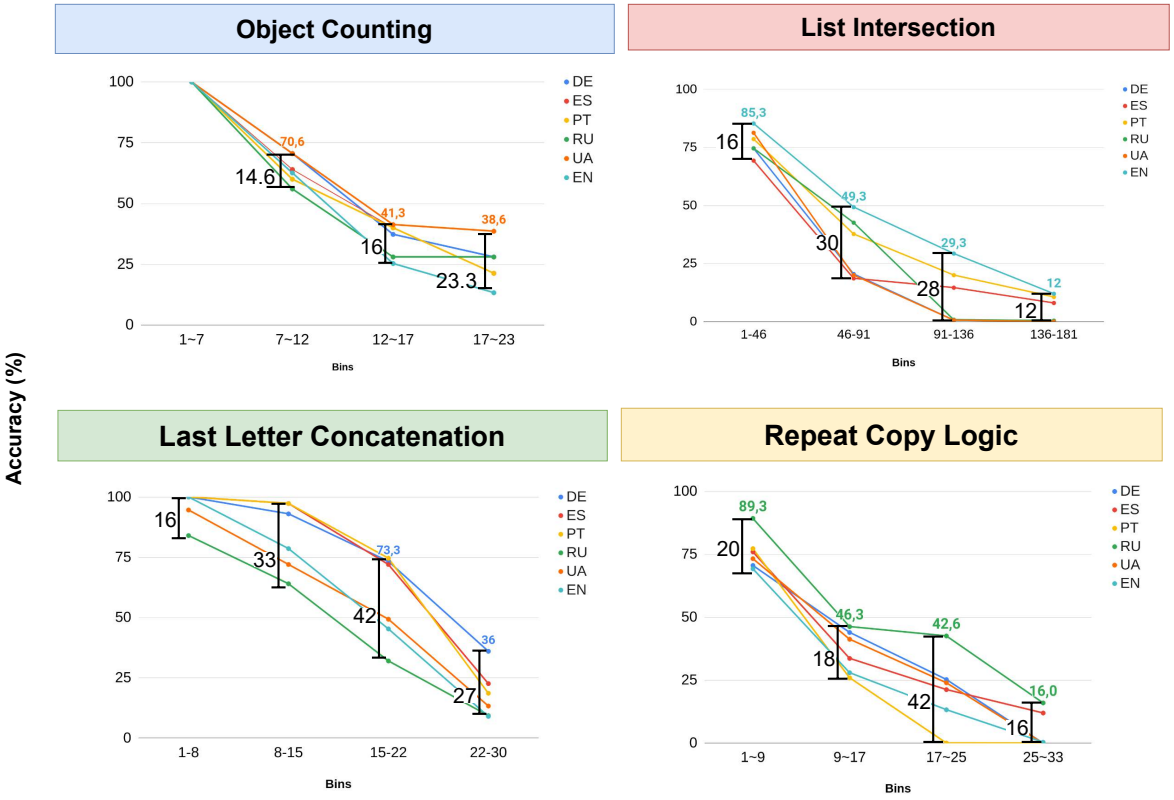

Figure 3: GPT-4 performance in the MLissard.

swered by GPT-4, which correctly answered 17 out of 32 original questions. We adopted this method to scale only the repetition factor; iii) Translating instructions using Google Translate and review the subset for accuracy; iv) Generate extrapolations on selected instructions, varying the repetition factor from 1 to 33 (see Table 1).

We randomly selected 15 of the 17 correctly answered questions for this phase.

4 Baseline Methods

The evaluation of each task involved analyzing responses from GPT-4 (gpt4-0613) and Llama-3 (Llama-3.1-405B-Instruct and Llama-3-instruction-70B) using greedy decoding. We observed no repetition issues. Each task was preceded by a predefined instruction (description of the task) with in-context examples: four for "Object Counting," "Find Intersection," and "Last Letter Concat," and one for "Repeat Copy Logic" because inputs already provided sufficient information to perform the task. Both the instructions and examples were in the target language of the evaluation. For instance, English tasks used English instructions and examples (see Figure 2 (a)). For the in-context examples used during model evaluation, we selected samples contained in the first bin, as these contain the smallest lengths.

We utilized the exact match as the primary metric. This methodology is further modified in section 5.2, where we discuss the impact of cross-language inputs on model performance.

5 Results

Figure 3 presents the results obtained via GPT-4 in the target tasks and languages. Overall, there is a gradual decline in the performance of language models across tasks as complexity increases, as measured by the number of key entities in the input sequence. For instance, in the "Object Counting" task, when presented with inputs containing 1 to 7 objects, the model achieve approximately 100% accuracy. However, their accuracy drops below 50% when confronted with sequences with 12 to 17 objects. This behavior is reflected in the target languages as well, all of which present a loss of more than 50% when dealing with more complex input sequences.

We also observed considerable variability in performance between languages depending on the spe-

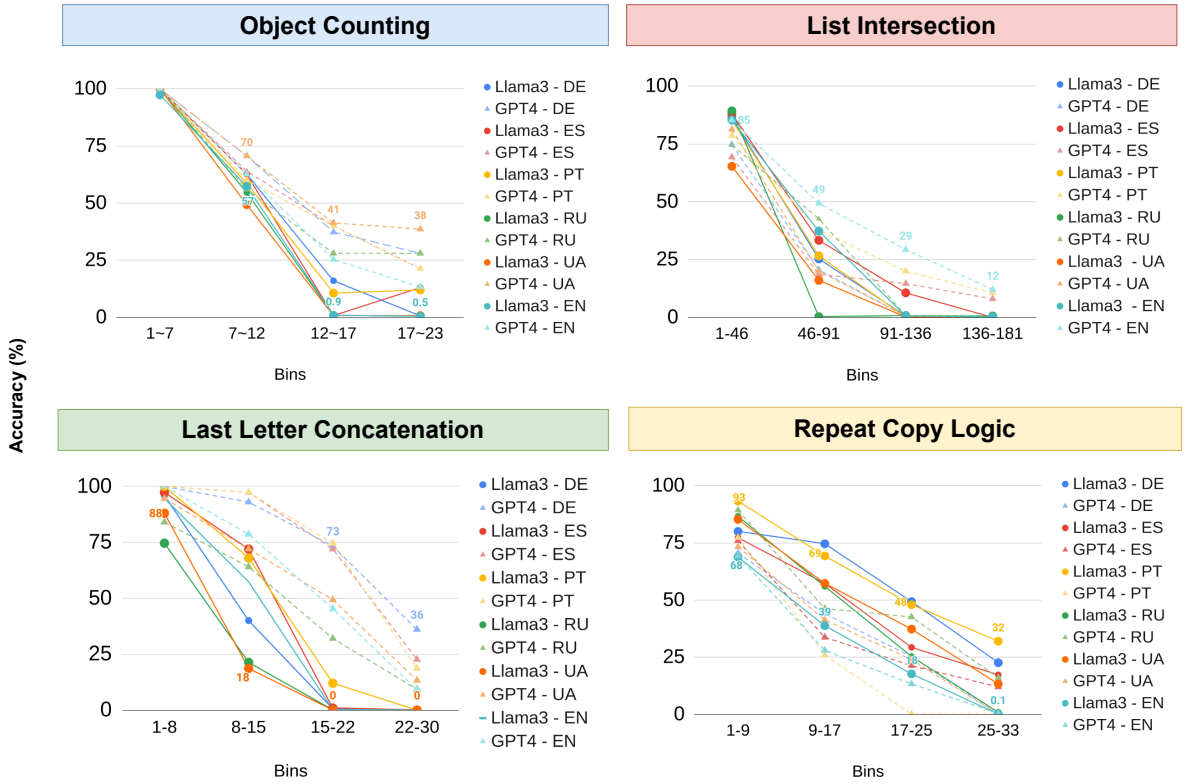

Figure 4: Comparison of Llama-3.1-405B vs. GPT-4 performance in the MLissard Benchmark

cific task. For instance, differences ranging from 2.4 to 42 points are observed in the intermediate bins for tasks such as "Last Letter Concatenation" and "Repeat Copy Logic". These variations are intriguing as there doesn't appear to be a general language preference. For example, in the "Last Letter Concatenation" task, German, Portuguese, and Spanish outperform Russian by a margin of 42.6 points in the 15-22 bin. Conversely, in the "Repeat Copy Logic" task, Russian outperforms Portuguese by 42.5 points.

Contrary to the general trend observed in studies of multilingual models, English did not exhibit exceptional performance when compared to other languages. Except for the "List Intersection" task, English consistently remained at an average or lower accuracy level across bins.

Generalization performance also varies between tasks; as demonstrated in Table 3, GPT-4 has greater difficulty executing the "List Intersection" and "Repeat Copy Logic" tasks. In the "List Intersection" task, the model achieves less than 10% accuracy in bins 3 and 4. In the "Repeat Copy Logic" task, accuracy drops to below 25% in the same bins. Both tasks require extensive memorization and state tracking. We hypothesize that these challenges, along with the increased sentence length, have influenced the observed performance outcomes.

Regarding the performance of open-source models in the MLissard benchmark, Figure 4 illustrates that both models performed similarly in bin 1, with accuracy points ranging between 70 and 100. However, as task complexity increased from bin 2 onwards, differences in performance stood out. Except for the "Repeat Copy Logic" task, GPT-4 outperformed Llama-3.1-405B by 5 to 60 accuracy points (see Table 3).

On the other hand, in the "Repeat Copy Logic" task, there is a reverse comparison, where Llama-3.1-405B outperforms GPT-4 in all bins, with the difference ranging from 9 points to 16 points of accuracy.

In relation to language preference behavior, both the Llama-3.1-405B and GPT-4 models exhibit similar task-dependent variations. Llama-3.1-405B demonstrates more consistent performance across Portuguese, German, and English.

5.1 Impact of model size

The Llama-3.1-405B model achieved state-of-the-art results in general NLP task benchmarks com-

Task	Bin 1		Bin 2		Bin 3		Bin 4	
	Llama	**GPT-4**	**Llama**	**GPT-4**	**Llama**	**GPT-4**	**Llama**	**GPT-4**
OC	100	100	58	**63**	0.8	**38**	0.7	**24.6**
LI	**86**	76	26	**29**	0.6	**7.7**	0.1	**4**
LLC	95	**100**	48.6	**85.8**	0.4	**60**	0	**16**
RCL	**82**	73.3	**57**	41.3	**33**	24	**15**	0.4
AVG	90.7	87	47.4	54.7	8.7	32.4	3.9	11.7

Table 3: Average accuracy of all languages per bin on tasks Object Counting (OC), List Intersection (LI), Last Letter Concatenation (LLC), and Repeat Copy Logic (RCL). Comparative result between the Llama-3.1-405B and GPT-4 models, highlighting in bold the best system performance in each bin.

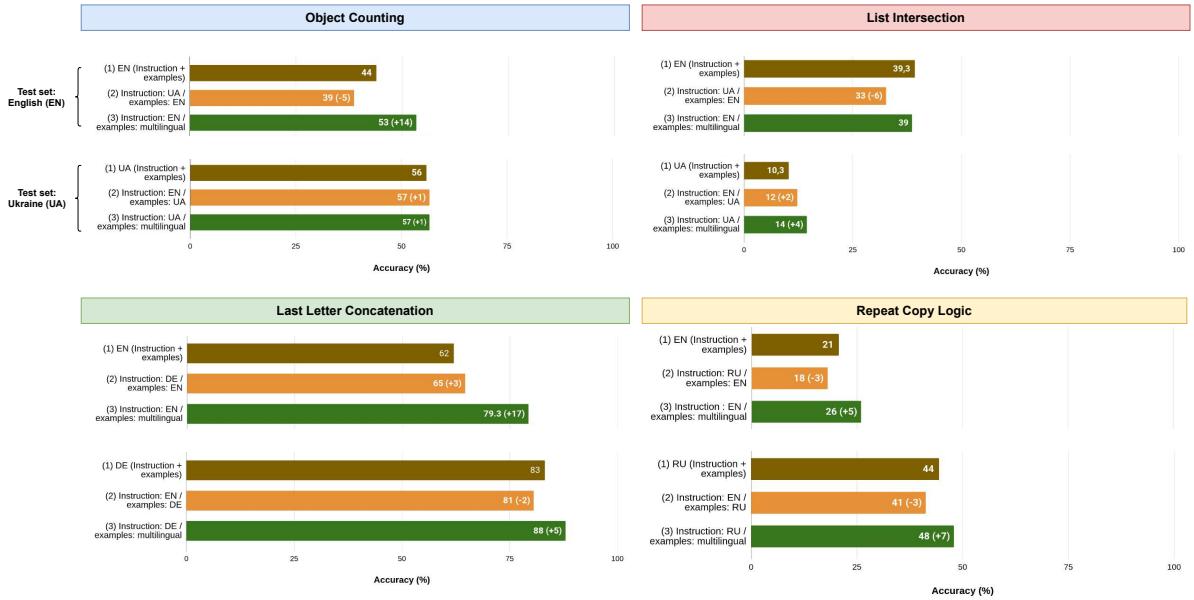

Figure 5: Average accuracy considering all bins. Since (1) Baseline - Both the instruction and the examples derive from the same target language; (2) instruction in the language that performed better or worse and a examples in the target language; (3) Instruction in target language and multilingual examples.

pared to the Llama-3-70B model. We investigated whether this performance trend is also evident in the MLissard benchmarks, especially in relation to the complexity indicated by the bins.

Table 4 compares the average performance of each bin (for all MLissard tasks) using the Llama-3.1-405B and Llama-3-70B models. As expected, Llama-3.1-405B significantly outperforms Llama-3-70B across all languages and complexity bins. The largest differences between the models occur in bins 1 and 2, with performance gaps ranging from 16 to 43 points. In contrast, for bins 3 and 4, which involve more complex tasks, the performance improvement is less pronounced, with variations ranging from 0.3 to 11 points. This suggests that Llama-3.1-405B, like the 70B version, also struggles with long sequences.

5.2 Can cross language improve extrapolation performance?

We aim to examine the impact on extrapolation performance by focusing on two components: 1) providing instructions in a different language than the target language, and 2) using mixed-language few-shot examples (see Figure 2 - (b)). For in-context examples, we used Portuguese, German, Ukrainian, and English. For the "Repeat Copy Logic" task, we provided two contextualized examples (English and Ukrainian), while for the other tasks, we provided four examples.

We conducted ablation tests on all tasks in the MLissard dataset using the GPT-4 model. For com-

Lang	Bin 1		Bin 2		Bin 3		Bin 4	
	70B	**405B**	**70B**	**405B**	**70B**	**405B**	**70B**	**405B**
EN	70.6	90	18.6	48	0.1	0.7	0	0.1
PT	79.3	96.6	24	63.3	0.1	11.3	0	6
ES	74	92.6	16.6	60	0.1	5.7	0	6.5
DE	74.6	91.3	16.8	51.3	0.5	8.3	0	0.3
RU	60.6	88	12.2	38	0	0.8	0	0.6
UA	55.3	86.6	10.7	33.9	0.1	0.5	0	0.4

Table 4: Average accuracy across all MLissard tasks was compared between the Llama-3-70B and Llama-3.1-405B models.

parative purposes, we focused on the languages that achieved the highest and lowest performance in each task. We then compared these results with the baseline (both instructions and examples in the same language).

Figure 5 presents the experimental results for each task. As shown in the results, when we gave prompts in a language different from the test set, accuracy declined by an average of 2.3 percentage points. However, when we kept instructions in the test target language but included paraphrased examples contextualized in multiple languages, performance improved by an average of 6.25 percentage points. This improvement ranged from 2 points in the "List Intersection" task to 17 points in the "Last Letter Concatenation" task and remained consistent across all evaluated languages. These findings indicate that contextual examples in multiple languages can improve the quality of extrapolation.

6 Conclusion

We presented a multilingual benchmark to evaluate the ability of language models to deal with long texts across languages. Our approach distinguishes itself from existing benchmarks through the introduction of a control mechanism, which we refer to as "key entities." This mechanism enables us to systematically increase task complexity in tandem with sequence length. Furthermore, the ability to solve these tasks is predicated on the repeated application of simple rules, providing more control and enabling a detailed analysis of model performance in relation to the frequency of rule application. This contrasts with benchmarks that rely on lengthy natural language texts, where the relationship between text length and task difficulty may become obscured. Despite the apparent simplicity of these tasks, they reveal significant limitations in state-of-the-art LLMs concerning the processing and generation of text as lengths increase. Our findings indicate that language and model size significantly affect extrapolation results. Moreover, including in-context examples in multiple languages improves MLissard's generalization performance.

7 Limitations

Our evaluations were conducted on a set of six languages, therefore, the findings of this work may not necessarily extend to other languages, particularly low-resource ones. Additionally, we solely employed a standard prompt style for our evaluations, and the performance with more sophisticated techniques, such as chain-of-thought (CoT) prompting, remains to be investigated. Finally, given the limitation of our study to two models (GPT-4 and Llama-3), the results may not generalize to other LLMs.

References

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Maxamed Axmed, Kalika Bali, and Sunayana Sitaram. 2023. Mega: Multilingual evaluation of generative ai. *Preprint*, arXiv:2303.12528.

Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. 2022. Exploring length generalization in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 38546–38556. Curran Associates, Inc.

BIG bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Mirelle Candida Bueno, Carlos Gemmell, Jeff Dalton, Roberto Lotufo, and Rodrigo Nogueira. 2022. Induced natural language rationales and interleaved markup tokens enable extrapolation in large language models. In *Proceedings of the 1st Workshop on Mathematical Natural Language Processing (MathNLP)*, pages 17–24, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional interpolation. *Preprint*, arXiv:2306.15595.

Ta-Chung Chi, Ting-Han Fan, Peter J Ramadge, and Alexander Rudnicky. 2022. Kerple: Kernelized relative positional embedding for length extrapolation. In *Advances in Neural Information Processing Systems*, volume 35, pages 8386–8399. Curran Associates, Inc.

Ta-Chung Chi, Ting-Han Fan, Alexander Rudnicky, and Peter Ramadge. 2023. Dissecting transformer length extrapolation via the lens of receptive field analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13522–13537, Toronto, Canada. Association for Computational Linguistics.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and Pedro A. Ortega. 2023. Neural networks and the chomsky hierarchy. *Preprint*, arXiv:2207.02098.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yann Dubois, Gautier Dagan, Dieuwke Hupkes, and Elia Bruni. 2020. Location Attention for Extrapolation to Longer Sequences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 403–413, Online. Association for Computational Linguistics.

Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2024. LM-infinite: Zero-shot extreme length generalization for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3991–4008, Mexico City, Mexico. Association for Computational Linguistics.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. 2023. The impact of positional encoding on length generalization in transformers. In *Advances in Neural Information Processing Systems*, volume 36, pages 24892–24928. Curran Associates, Inc.

Guolin Ke, Di He, and Tie-Yan Liu. 2021. Rethinking positional encoding in language pre-training. In *International Conference on Learning Representations*.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. 2019. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 3843–3857. Curran Associates, Inc.

Changmao Li and Jeffrey Flanigan. 2024. Task contamination: Language models may not be few-shot anymore. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):18471–18480.

Dacheng Li, Rulin Shao, Anze Xie, Ying Sheng, Lianmin Zheng, Joseph E. Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. 2023a. How long can open-source llms truly promise on context length?

Shanda Li, Chong You, Guru Guruganesh, Joshua Ainslie, Santiago Ontanon, Manzil Zaheer, Sumit Sanghai, Yiming Yang, Sanjiv Kumar, and Srinadh Bhojanapalli. 2023b. Functional interpolation for relative positions improves long context transformers. *Preprint*, arXiv:2310.04418.

Adam Liška, Germán Kruszewski, and Marco Baroni. 2018. Memorize or generalize? searching for a compositional rnn in a haystack. *arXiv preprint arXiv:1802.06467*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2021. Investigating the limitations of transformers with simple arithmetic tasks. *arXiv preprint arXiv:2102.13019*.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena.

2021. Show your work: Scratchpads for intermediate computation with language models. *Preprint*, arXiv:2112.00114.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Ofir Press, Noah Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*.

Zhen Qin, Yiran Zhong, and Hui Deng. 2023. Exploring transformer extrapolation. *Preprint*, arXiv:2307.10156.

Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. 2023. ZeroSCROLLS: A zero-shot benchmark for long text understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7977–7989, Singapore. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Sean Welleck, Peter West, Jize Cao, and Yejin Choi. 2022. Symbolic brittleness in sequence models: on systematic generalization in symbolic mathematics. In *AAAI*.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024. ∞bench: Extending long context evaluation beyond 100k tokens. *Preprint*, arXiv:2402.13718.

Liang Zhao, Xiaocheng Feng, Xiachong Feng, Bin Qin, and Ting Liu. 2023. Length extrapolation of transformers: A survey from the perspective of position encoding. *Preprint*, arXiv:2312.17044.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023a. Least-to-most prompting enables complex reasoning in large language models. *Preprint*, arXiv:2205.10625.

Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Josh Susskind, Samy Bengio, and Preetum Nakkiran. 2023b. What algorithms can transformers learn? a study in length generalization. In *ICLR, NeurIPS Workshop*.