UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Mirelle Candida Bueno

MLissard: Benchmarks de raciocínio sequencial simples multilíngue

MLissard: Multilingual Long and Simple Sequential Reasoning Benchmarks

Campinas
2025

Mirelle Candida Bueno

MLissard:
Multilingual Long and Simple Sequential Reasoning Benchmarks

MLissard:
Benchmarks de raciocínio sequencial simples multilíngue

Dissertation presented to the School of Electrical and Computer Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Electrical Engineering, the area of Computer Engineering.

Dissertação de mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestra em Engenharia Elétrica, na área de Engenharia de Computação.

Orientador: Roberto de Alencar Lotufo

Este trabalho corresponde à versão final da dissertação defendida pela aluna Mirelle Candida Bueno, e orientada pelo Prof. Dr. Roberto de Alencar Lotufo.

Campinas
2025

Ficha catalográfica
Universidade Estadual de Campinas (UNICAMP)
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

B862m

Bueno, Mirelle Candida, 1997-
MLissard : multilingual long and simple sequential reasoning benchmarks / Mirelle Candida Bueno. – Campinas, SP : [s.n.], 2025.

Orientador: Roberto de Alencar Lotufo.
Dissertação (mestrado) – Universidade Estadual de Campinas (UNICAMP), Faculdade de Engenharia Elétrica e de Computação.

1. Processamento de linguagem natural. 2. Aprendizado profundo. 3. Inteligência artificial. I. Lotufo, Roberto de Alencar, 1955-. II. Universidade Estadual de Campinas (UNICAMP). Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações complementares

Título em outro idioma: MLissard : benchmarks de raciocínio sequencial simples multilíngue
Palavras-chave em inglês:
Natural language processing
Deep learning
Artificial intelligence
Área de concentração: Engenharia de Computação
Titulação: Mestra em Engenharia Elétrica
Banca examinadora:
Roberto de Alencar Lotufo. [Orientador]
Thiago Alexandre Salgueiro Pardo
Jayr Alencar Pereira
Data de defesa: 20-01-2025
Programa de Pós-Graduação: Engenharia Elétrica

Objetivos de Desenvolvimento Sustentável (ODS)
Não se aplica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0003-2374-6123
- Currículo Lattes do autor: http://lattes.cnpq.br/1399710717515695

Prof. Dr. Roberto de Alencar Lotufo (Presidente)
Prof. Dr. Thiago Alexandre Salgueiro Pardo
Prof. Dr. Jayr Alencar Pereira

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

Acknowledgement

The successful completion of my master's degree was made possible through the immeasurable support of incredible people who accompanied me on this journey.

I am profoundly grateful to God and my family for their unwavering encouragement and to my fiancé, Eduardo Ferreira, who stood by me at every stage, offering motivation and trust.

I extend my heartfelt thanks to my advisors, Roberto Lotufo and Rodrigo Nogueira, whose guidance, teachings, and insights were invaluable throughout the process. Their mentorship made the conception and completion of this project possible.

I am also deeply appreciative of my colleagues in the postgraduate program for their stimulating discussions and idea exchanges, as well as my friends who, directly or indirectly, supported and assisted me in developing this project.

Lastly, a special thank you goes to UNICAMP and the postgraduate coordination team at the School of Electrical and Computer Engineering (FEEC) for their steadfast support and guidance.

Resumo

Os modelos de linguagem agora são capazes de resolver tarefas que exigem lidar com longas sequências consistindo em centenas de milhares de tokens. No entanto, eles frequentemente falham em tarefas que exigem o uso repetitivo de regras simples, mesmo em sequências que são muito mais curtas do que aquelas vistas durante o treinamento. Por exemplo, grandes modelos de linguagem (LLMs) de última geração podem encontrar itens comuns em duas listas com até 20 itens, mas falham quando as listas têm 80 itens.

Esta dissertação apresenta o MLissard, um benchmark multilíngue projetado para avaliar as habilidades dos modelos de processar e gerar textos de tamanhos variados e oferece um mecanismo para controlar a complexidade da sequência. Os resultados demostraram que tanto os modelos de código aberto e proprietários apresentaram um declínio consistente no desempenho em todos as tarefas e idiomas à medida que a complexidade da sequência aumenta. Surpreendentemente, o uso de exemplos em contexto em idiomas diferentes do inglês ajuda a aumentar significativamente o desempenho da extrapolação.

Palavras-chave: Processamento de Linguagem Natural; Transformers; Aprendizado de Máquina; Modelo de Linguagem de Grande Escala; Extrapolação.

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 large language models (LLMs) can find common items in two lists with up to 20 items but fail when lists have 80 items.

This paper introduces MLissard, a multilingual benchmark specifically designed to assess the performance of models in processing and generating texts of varying lengths, while also providing a mechanism to control sequence complexity. The results demonstrate that both open-source and proprietary models show a consistent decline in performance across all tasks 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.

Keywords: Natural Language Processing; Transformers; Machine learning; Large Language Model; Extrapolation.

List of Figures

List of Tables

Contents

1 Introduction

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, Anil et al., 2022, 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].

In practical applications, this challenge is evident in LLM-based assistants, particularly when they encounter requests necessitating the use of simple, repetitive rules or memorization. For instance, tasks such as identifying a specific name within an extensive contact list or checking priorities in a task list exemplify this issue; the scope of similar tasks is vast. Consequently, various techniques have been developed to address these challenges, both at the architectural level of transformers with an emphasis on positional embeddings and at the prompt level by breaking tasks into sequential steps.

Regardless of the approaches implemented, there is still a lack of evaluation benchmarks for subsequent tasks. Benchmarks such as SCROLLS Shaham et al. [2022] assess models on long sequence tasks but lack explicit control over task complexity relative to sequence length, making it hard to gauge length generalization. For example, there is no certainty that questions about longer texts are harder than those about shorter ones. This highlights the need for benchmarks that explicitly test the impact of sequence length. In dialogue Li et al. [2023a] and multi-document question answering Liu et al. [2024], techniques like retrieval-augmented generation (RAG) are prevalent, and therefore explicitly isolating the length extrapolation issue poses a challenge.

We introduce MLissard, a multilingual benchmark that evaluates models on tasks with repetitive rules, where difficulty increases with sequence length. Supporting English (EN), German (DE), Portuguese (PT), Russian (RU), Spanish (ES), and Ukrainian (UA), MLissard employs a developed method to identify models' breaking points. Additionally, it allows for the generation of new examples to heighten task difficulty, thereby avoiding the contamination issues commonly associated with traditional datasets Ahuja et al. [2023], Li and Flanigan [2024]. At the time of this research, it is the first multilingual dataset aimed at assessing models' length extrapolation capabilities.

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. Our findings underscore that irrespective of their architectures and parameter counts, all examined models demonstrate a performance degradation with in-

creasing length, controlled by the number of key entities (see their definition in Table 1), 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 investigation 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.

The subsequent sections will explore the primary contributions this article makes to the community regarding the topic of extrapolation. We will examine related works that discuss approaches and techniques in the literature aimed at enhancing the generalization capacity of language models, particularly in addressing the challenges related to sentence length. The Methodology section outlines the development of benchmarks and details the key entities method, along with the various tasks involved. The Experimental Setup section describes the process of evaluating the MLissard using state-of-the-art language models. In the Results section, we present the significant findings from the experimentation phase and the ablation tests. Lastly, the Conclusion and Limitations section reviews the results and discusses the challenges encountered in completing the master's thesis work.

1.1 Main Contributions

The developed thesis makes contributions in the field of language model generalization, including:

1. The creation of a multilingual benchmark that can be easily expanded with new examples, lengths, and languages.

2. Identification of model breaking points through a key entity mechanism, aiding in tracking model improvements in extrapolation tasks.

3. An analysis of the impact of language diversity and model size on extrapolation capabilities.

4. Provision of open-source datasets and results.

Additionally, two papers related to the master's thesis were accepted for publication: one at the Math NLP workshop during EMNLP-2022, which has been cited 14 times, and another at the Genbench workshop during EMNLP-2024 that was cited two times. Both publications significantly contributed to the development of the dissertation topic by highlighting the limitations of neural architectures, particularly their fragility in extrapolating internal rules—a capability that is not typically emphasized during the pre-training phase.

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. [2021] 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. [2022] and Zhou et al. [2023a] 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 last-letter concatenation (symbolic manipulation), SCAN Lake and Baroni [2018] (compositional generalization), and mathematical reasoning. The last letter concatenation task focuses on string manipulation by requiring the concatenation of the last letters of each word. In contrast, the SCAN task involves using a limited vocabulary (e.g., JUMP, TWICE, LEFT, RIGHT, etc.) alongside an instruction, such as JUMP TWICE, which the model must translate into a sequence of navigation commands like JUMP JUMP. This task effectively evaluates the model's ability to apply a repetitive rule and combine elements to produce the correct output. Additionally, Bueno et al. [2022] 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. [2024] 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. [2023] evaluated commonly used positional coding methods and found that, despite these methods improving perplexity score, completely omitting positional coding actually yielded better results on downstream tasks requiring extrapolation ability.

The studies above provide evidence of multiple approaches that have been developed to address the challenge of extrapolation. Nonetheless, there is a notable absence of research focusing on development of diverse and standardized datasets that assess the generation and synthesis of extended text sequences produced by neural models. This gap is particularly significant when considering that many of the classical datasets available in question may have already been used into the training of large language models.

3 Methodology

Our benchmark incorporates a combination of existing tasks, such as those from BIG- bench authors [2023], as well as newly developed ones. We intentionally excluded classical datasets (e.g., SCAN) from the analysis, since unlike the synthetic MLissard, their test sets are publicly available and many solutions have been extensively detailed in the scientific literature, making them potentially familiar to large language models (LLMs). To create the benchmarks, several steps were followed:

1. **Filtering and creating tasks:** In this step, we identify simple and synthetic tasks that require logical reasoning, memorization, and repetition, and that include an extrapolation factor, such as lists of objects or command applications. The goal is to isolate this extrapolation factor—like the number of objects in a list—and then expand the task synthetically to cover larger input lengths.

2. **Identifying key entities:** As explained in Section 3.1, we extracted the extrapolation factor for each task and identified the possible range values to assess the model's generalizability.

3. **Generating new examples:** For each task, we developed a Python script to generate new examples, increasing the complexity of the key entity within the specified ranges.

4. **Multilingual adaptation:** In addition to English (EN), our language set includes German (DE), Spanish (ES), Portuguese (PT), Russian (RU), and Ukrainian (UA). We expanded this set by integrating automatic translation systems and using Python scripts to generate synthetic data.

Once the evaluation benchmark was created, we conducted tests to assess the extrapolation performance of leading state-of-the-art models. The following sections provide a detailed explanation of the concept of key entities, the generation process for each task, and the methodology used to evaluate the language models in MLissard.

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 performed empirically to achieve a balanced range

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 1: Key task entities: Last Letter Concatenation (LLC), Repeat Copy Logic (RCL), Object Counting (OC), and List Intersection (LI)

of difficulty levels. Bin 1 consists of sequences of shorter length, while Bin 4 comprises sequences of longer length. Table 1 describes the key entities and the respective lengths in each bin.

3.2 Tasks

In total, four tasks were developed, and Table 2 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.

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 2: Task Summary in the MLissard Benchmark.

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 3. 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.

Input	Target	Language
I have three onions, two potatoes, and a cabbage.	6	English
Ich habe einen Blumenkohl, eine Kartoffel, einen Kohl, einen Knoblauch, eine Yamswurzel, einen Salatkopf, eine Karotte, zwei Stangen Sellerie, vier Brokkoliköpfe und eine Zwiebel.	14	German

Table 3: Examples of input and target sequences of the Object Counting task.

3.2.2 List Intersection

The objective of this task is to find common items in two lists as exemplified in Table 4. 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. The methodology involved selecting the most frequent words because the language models had likely encountered them during pre-training. This ensured that the vocabulary used in the task would not pose any additional difficulty. 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.

Input	Target	Language
A: messed/sin/college/paul B: college/tough/alert/finger	college	English
A: disparar/arte/adiós/diste B: diste/arte/decirnos/bastante	arte diste	Spanish

Table 4: Examples of input and target sequences of the List Intersection task.

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 within an input

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

sequence comprised of random names. Table 5 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 of "List Intersection" task; 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.

Input	**Target**	**Language**
Noah Miles Emilia	h s a	English
Luis Marcia Pedro Marcelo Fernanda Maria	s a o o a a	Portuguese

Table 5: Examples of input and target sequences of the Last Letter Concatenation task.

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. Our methodology for creating the dataset includes:

1. Collecting responses to all input sequences from the BIG-bench repository[3];

2. Filtering responses to retain only those correctly answered by GPT-4, which correctly answered 17 out of 32 original questions. We adopted this methodology to focus on assessing the model's ability to handle extrapolation. Consequently, we selected easier questions and synthetically recreated them by varying the required repetitions in the task.

3. Translating instructions using Google Translate to support multiple languages and reviewing the subset for accuracy;

4. Generate extrapolations on selected instructions, varying the repetition factor (see Table 6).

[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/`

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

Input	**Target**	**Language**
Repeat 4 times "hello world"	hello world hello world hello world hello world	English
Repeat the word dog 10 times, but halfway through also say woof	dog dog dog dog dog woof dog dog dog dog dog	English
Say the days of the week but only the weekend days, 2 times	Saturday Sunday Saturday Sunday	English
Wiederhole „Hallo Welt“ 5 Mal	hallo welt hallo welt hallo welt hallo welt hallo welt	German

Table 6: Examples of input and target sequences of the Repeat Copy Logic task.

4 Experimental Setup

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 1 (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 metric is determined by dividing the number of correctly answered examples by the total number of examples evaluated. The percentage range 0-100% is returned by the metric. This methodology is further modified in section 5.2, where we discuss the impact of cross-language inputs on model performance.

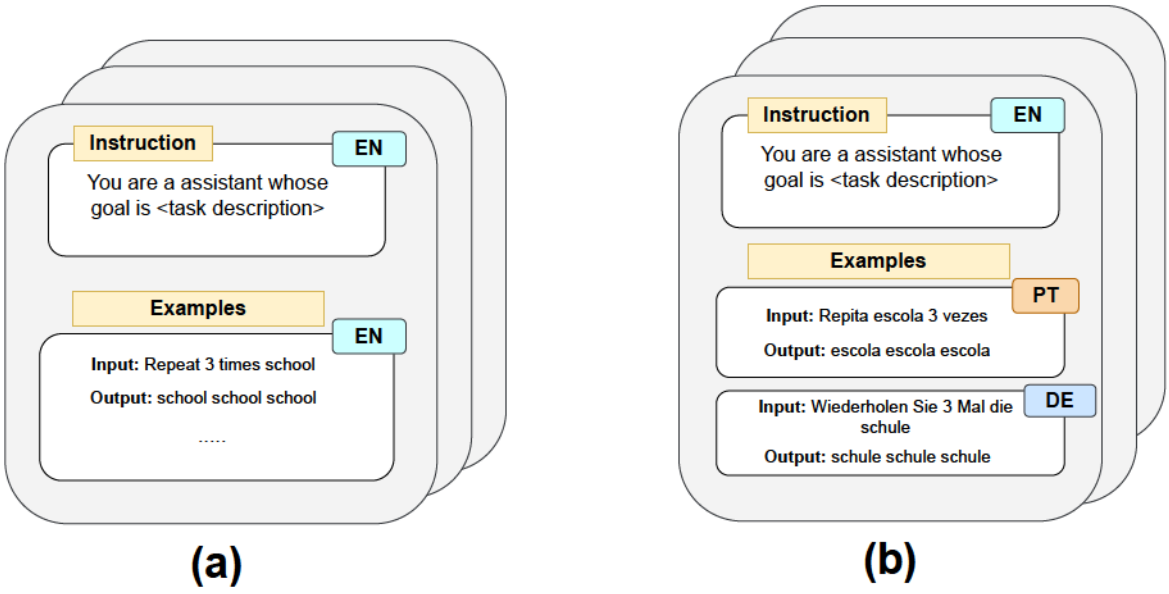

Figure 1: Template for evaluation. Being (a) Instruction and examples of tasks in the target language; (b) Instruction in the target language and multilingual examples.

5 Results

Figure 2 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.

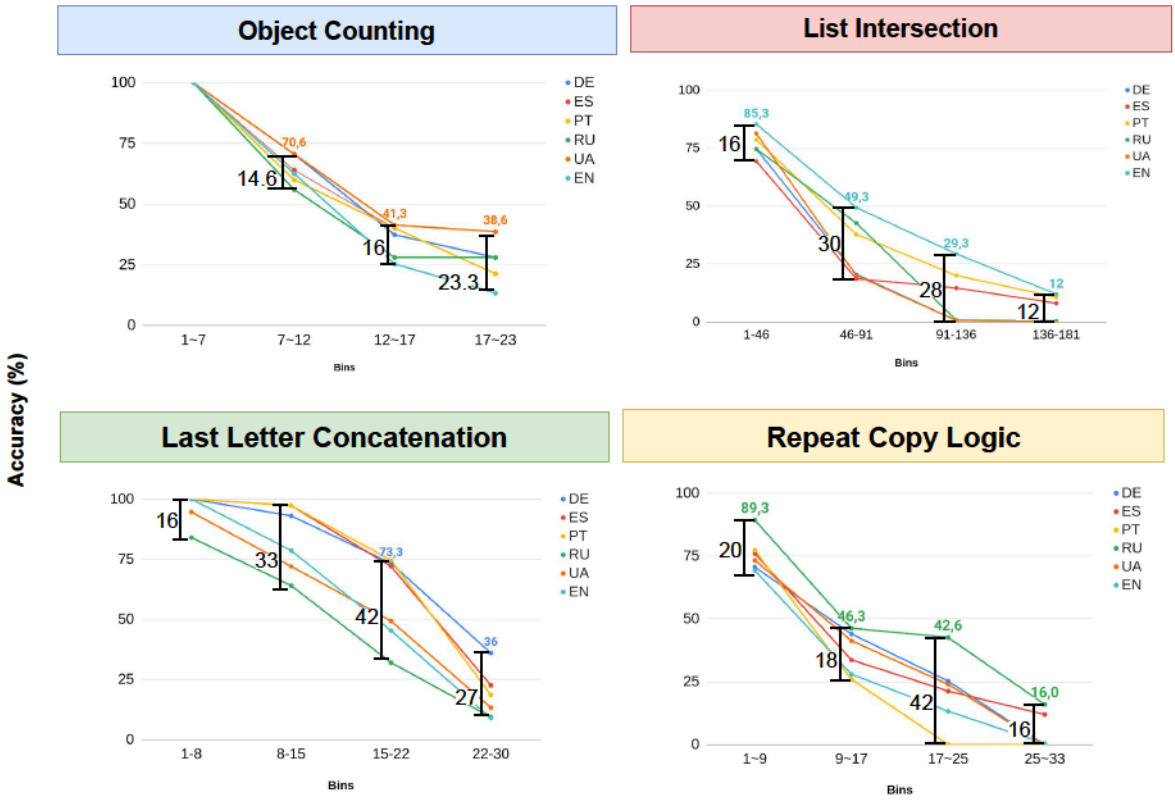

Figure 2: GPT-4 performance in the MLissard.

We also observed considerable variability in performance between languages depending on the specific 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 compared to other languages. In fact, except for

the "List Intersection" task, English consistently demonstrated average or below-average accuracy across bins. This pattern may be attributed to the nature of the MLissard tasks, which prioritize reasoning and memorization for extrapolation over advanced language knowledge.

The intrinsic nature of the tasks also significantly impacts generalization performance. As demonstrated in Table 7, 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 3 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 7).

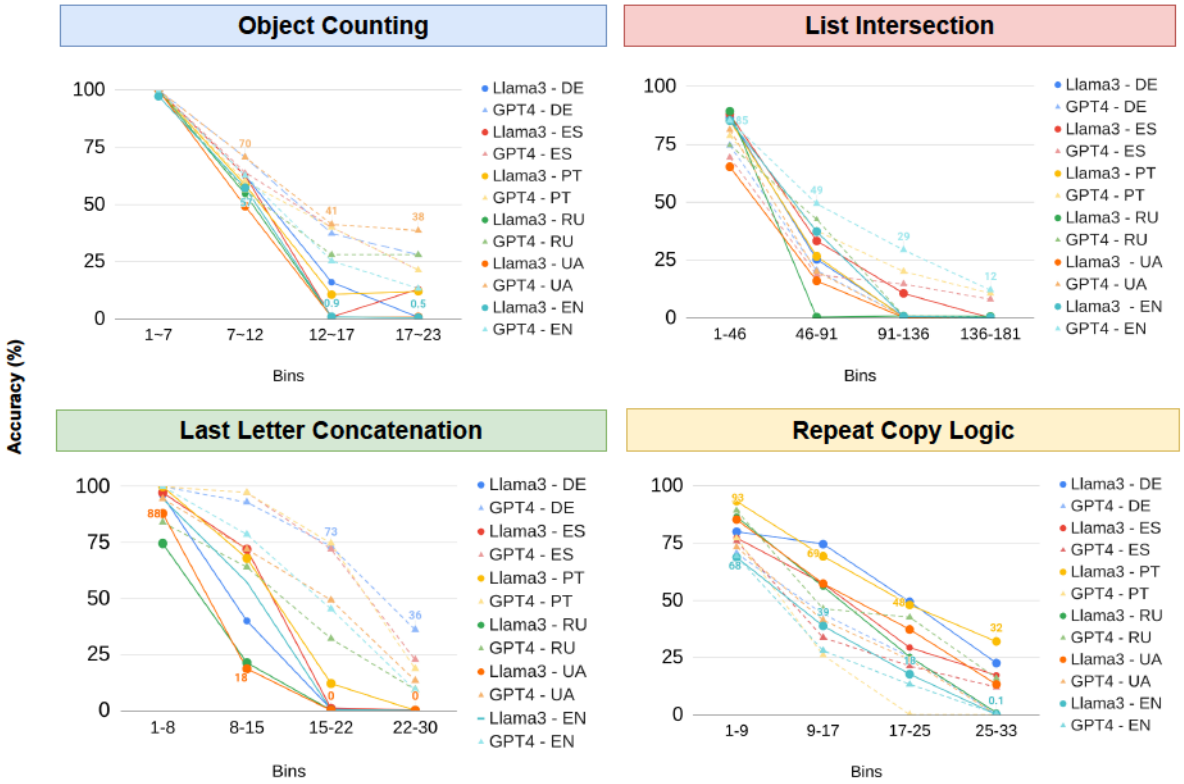

Figure 3: Comparison of Llama-3.1-405B vs. GPT-4 performance in the MLissard Benchmark

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.

Task	**Bin 1**		**Bin 2**		**Bin 3**		**Bin 4**	
	Llama	**GPT-4**	**Llama**	**GPT-4**	**Llama**	**GPT-4**	**Llama**	**GPT-4**
OC	100.0	100.0	57.9	**63.3**	0.85	**38.6**	0.70	**24.6**
LI	**85.9**	76.6	25.9	**29.1**	0.60	**7.70**	0.15	**4.2**
LLC	95.3	**100.0**	48.6	**85.8**	0.40	**60.6**	0.0	**15.9**
RCL	**82.6**	73.3	**57.3**	41.3	**33.3**	24.0	**15.3**	0.40
AVG	**90.9**	87.4	47.4	**54.8**	8.7	**32.7**	3.5	**11.2**

Table 7: The average accuracy for each language across the tasks of Object Counting (OC), List Intersection (LI), Last Letter Concatenation (LLC), and Repeat Copy Logic (RCL) is presented. We compare the performance of two models: Llama-3.1-405B and GPT-4, with the best results highlighted in bold.

5.1 Impact of model size

The Llama-3.1-405B model achieved state-of-the-art results in general NLP task benchmarks compared 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 8 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 1 - (b)). For the test with mixed languages, we used examples in Portuguese, German, Ukrainian, and English. These languages showed greater performance in MLissard with the default prompt; therefore, we

Lang	Bin 1		Bin 2		Bin 3		Bin 4	
	70B	**405B**	**70B**	**405B**	**70B**	**405B**	**70B**	**405B**
EN	70.6	89.9	18.6	48.0	0.15	0.75	0.0	0.15
PT	79.3	96.6	23.9	63.3	0.0	11.3	0.0	6.1
ES	73.9	92.6	16.6	59.9	0.1	5.7	0.0	6.5
DE	74.6	91.3	16.8	51.3	0.05	8.3	0.0	0.3
RU	60.6	87.9	12.2	37.9	0.0	0.8	0.0	0.6
UA	55.3	86.6	10.7	33.9	0.0	0.5	0.0	0.4

Table 8: Average accuracy across all MLissard tasks was compared between the Llama-3-70B and Llama-3.1-405B models.

would like to test if mixing them can increase the extrapolation performance in general languages. 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 comparative 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 4 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.

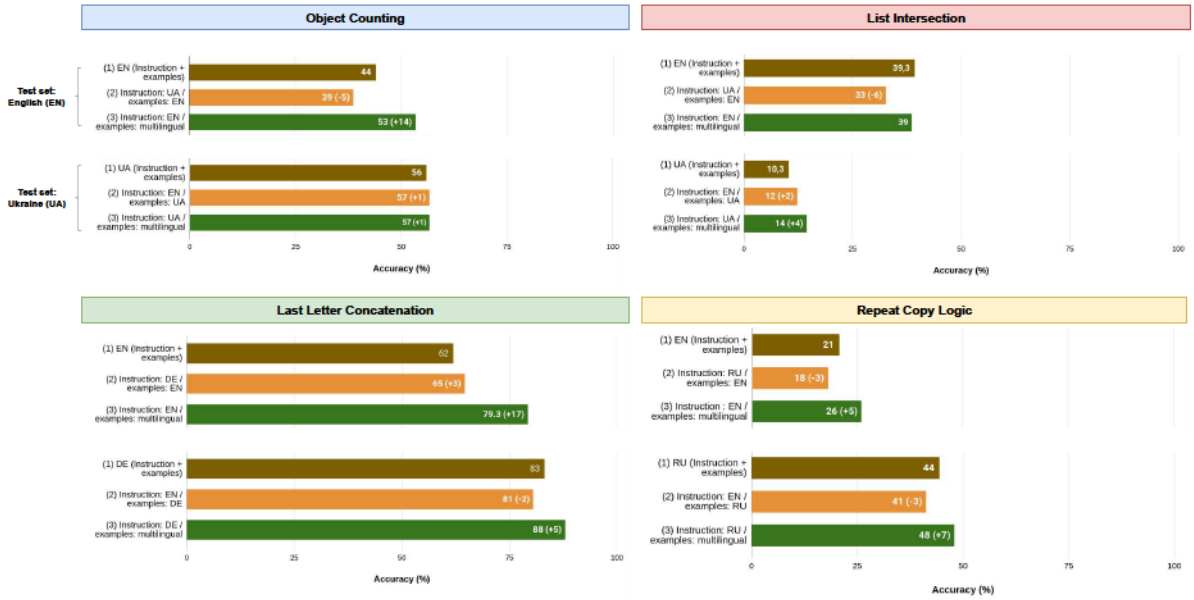

Figure 4: Average accuracy considering all bins. Since (1) Baseline - Both the instruction and the examples derive from the same target language; (2) instruction in the language that performed better or worse and a examples in the target language; (3) Instruction in target language and multilingual examples.

6 Conclusion

This master's thesis introduces 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.

6.1 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

K. Ahuja, H. Diddee, R. Hada, M. Ochieng, K. Ramesh, P. Jain, A. Nambi, T. Ganu, S. Segal, M. Axmed, K. Bali, and S. Sitaram. Mega: Multilingual evaluation of generative ai, 2023.

C. Anil, Y. Wu, A. Andreassen, A. Lewkowycz, V. Misra, V. Ramasesh, A. Slone, G. Gur-Ari, E. Dyer, and B. Neyshabur. Exploring length generalization in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 38546–38556. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/fb7451e43f9c1c35b774bcfad7a5714b-Paper-Conference.pdf.

B. bench authors. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=uyTL5Bvosj.

M. C. Bueno, C. Gemmell, J. Dalton, R. Lotufo, and R. Nogueira. Induced natural language rationales and interleaved markup tokens enable extrapolation in large language models. In D. Ferreira, M. Valentino, A. Freitas, S. Welleck, and M. Schubotz, editors, *Proceedings of the 1st Workshop on Mathematical Natural Language Processing (MathNLP)*, pages 17–24, Abu Dhabi, United Arab Emirates (Hybrid), Dec. 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.mathnlp-1.3. URL https://aclanthology.org/2022.mathnlp-1.3.

S. Chen, S. Wong, L. Chen, and Y. Tian. Extending context window of large language models via positional interpolation, 2023. URL https://arxiv.org/abs/2306.15595.

T.-C. Chi, T.-H. Fan, P. J. Ramadge, and A. Rudnicky. Kerple: Kernelized relative positional embedding for length extrapolation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 8386–8399. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/37a413841a614b5414b333585e7613b8-Paper-Conference.pdf.

T.-C. Chi, T.-H. Fan, A. Rudnicky, and P. Ramadge. 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, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.756. URL https://aclanthology.org/2023.acl-long.756.

G. Delétang, A. Ruoss, J. Grau-Moya, T. Genewein, L. K. Wenliang, E. Catt, C. Cundy, M. Hutter, S. Legg, J. Veness, and P. A. Ortega. Neural networks and the chomsky hierarchy, 2023.

A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Y. Dubois, G. Dagan, D. Hupkes, and E. Bruni. Location Attention for Extrapolation to Longer Sequences. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 403–413, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.39. URL https://aclanthology.org/2020.acl-main.39.

C. Han, Q. Wang, H. Peng, W. Xiong, Y. Chen, H. Ji, and S. Wang. LM-infinite: Zero-shot extreme length generalization for large language models. In K. Duh, H. Gomez, and S. Bethard, editors, *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, June 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.naacl-long.222.

A. Kazemnejad, I. Padhi, K. Natesan Ramamurthy, P. Das, and S. Reddy. The impact of positional encoding on length generalization in transformers. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 24892–24928. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/4e85362c02172c0c6567ce593122d31c-Paper-Conference.pdf.

D. Keysers, N. Schärli, N. Scales, H. Buisman, D. Furrer, S. Kashubin, N. Momchev, D. Sinopalnikov, L. Stafiniak, T. Tihon, et al. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*, 2019.

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

A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo, Y. Wu, B. Neyshabur, G. Gur-Ari, and V. Misra. Solving quantitative reasoning problems with language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 3843–3857. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/18abbeef8cfe9203fdf9053c9c4fe191-Paper-Conference.pdf.

C. Li and J. Flanigan. Task contamination: Language models may not be few-shot anymore. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):18471–18480, Mar. 2024. doi: 10.1609/aaai.v38i16.29808. URL https://ojs.aaai.org/index.php/AAAI/article/view/29808.

D. Li, R. Shao, A. Xie, Y. Sheng, L. Zheng, J. E. Gonzalez, I. Stoica, X. Ma, and H. Zhang. How long can open-source llms truly promise on context length?, 6 2023a. URL https://lmsys.org/blog/2023-06-29-longchat.

S. Li, C. You, G. Guruganesh, J. Ainslie, S. Ontanon, M. Zaheer, S. Sanghai, Y. Yang,

S. Kumar, and S. Bhojanapalli. Functional interpolation for relative positions improves long context transformers, 2023b.

A. Liška, G. Kruszewski, and M. Baroni. Memorize or generalize? searching for a compositional rnn in a haystack. *arXiv preprint arXiv:1802.06467*, 2018.

N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL `https://aclanthology.org/2024.tacl-1.9`.

R. Nogueira, Z. Jiang, and J. Lin. Investigating the limitations of transformers with simple arithmetic tasks. *arXiv preprint arXiv:2102.13019*, 2021.

M. Nye, A. J. Andreassen, G. Gur-Ari, H. Michalewski, J. Austin, D. Bieber, D. Dohan, A. Lewkowycz, M. Bosma, D. Luan, C. Sutton, and A. Odena. Show your work: Scratchpads for intermediate computation with language models, 2021. URL `https://arxiv.org/abs/2112.00114`.

OpenAI. Gpt-4 technical report, 2023.

O. Press, N. Smith, and M. Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=R8sQPpGCv0`.

Z. Qin, Y. Zhong, and H. Deng. Exploring transformer extrapolation, 2023.

U. Shaham, E. Segal, M. Ivgi, A. Efrat, O. Yoran, A. Haviv, A. Gupta, W. Xiong, M. Geva, J. Berant, and O. Levy. SCROLLS: Standardized CompaRison over long language sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12007–12021, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics. URL `https://aclanthology.org/2022.emnlp-main.823`.

J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf`.

S. Welleck, P. West, J. Cao, and Y. Choi. Symbolic brittleness in sequence models: on systematic generalization in symbolic mathematics. In *AAAI*, 2022. URL `https://arxiv.org/pdf/2109.13986.pdf`.

L. Zhao, X. Feng, X. Feng, B. Qin, and T. Liu. Length extrapolation of transformers: A survey from the perspective of position encoding, 2023.

D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le, and E. Chi. Least-to-most prompting enables complex reasoning in large language models, 2023a.

H. Zhou, A. Bradley, E. Littwin, N. Razin, O. Saremi, J. Susskind, S. Bengio, and

P. Nakkiran. What algorithms can transformers learn? a study in length generalization. In *ICLR, NeurIPS Workshop*, 2023b. URL https://arxiv.org/abs/2310.16028.