The Impact of Reasoning Methods across Languages

Anonymous ACL submission

Abstract

Previous works have been showing the effective operation of *reasoning methods* to elicit Large Language Models (LLMs) in delivering multi-step reasoned answers. Although these methods, best exemplified by Chain-of-Thought (CoT) and Program-Aided Language Models (PAL), have demonstrated actual functionalities in monolingual (English), multi-, and cross-lingual scenarios, are under-explored and lack an in-depth understanding. To address this gap, we propose a multidimensional analysis using five Cross-lingual tasks, experimenting with the impact of *reasoning methods* in different LLMs selected per families and scope of construction. Our results reveal that the effectiveness of *reasoning methods* varies significantly across models, tasks, and languages. In particular, higher-parameter LLMs, when elicited via CoT, are able to deliver reasoned multi-step answers better than smaller LLMs. In contrast, LLMs prompted via PAL achieve significant improvements anyway of the number of parameters. Finally, by analyzing the role of in-context cross-lingual demonstrations, we reveal that although they may provide benefits significantly in low-resource scenarios, their effectiveness is related to a proper trade-off between quantity and quality of demonstrations.

1 Introduction

One of the emergent properties of Large Language Models (LLMs) is the ability to solve tasks through prompts defined by structured patterns. This phenomenon, known as in-context learning (Brown et al., 2020; Dong et al., 2023), allows a task to be solved without updating the model parameters using only the input structure. In light of the success of in-context learning, there has been increased interest in better analyzing the factors that influence its operative, such as the selection of demonstrations (Liu et al., 2022; Rubin et al., 2022; Zhao et al., 2023) and prompts design (Zhang et al., 2022; Min et al., 2022a; Si et al., 2023).

In the case of *reasoning methods*, Chain-of-Thought (CoT) (Kojima et al., 2023; Wei et al., 2023), and Program-Aided Language Models (PAL) (Gao et al., 2022) have been emerged as two effective approaches. The first method, i.e., CoT, breaks down a reasoning problem into a series of intermediate steps using natural language, assembling it more general, flexible, and understandable. PAL offers solutions via Python functions, with its step-by-step programming code guaranteeing a more rigorous and structured methodology.

However, recent work investigates the operation of these methods mainly in English, while the study in multi and cross-lingual scenarios lags. This is problematic, as findings that apply to English may not be valid for other languages, especially those lower represented in LLMs training data.

Although there are some studies that go beyond English, they focus on benchmarking LLMs on natural language inference (Yang et al., 2019; Ponti et al., 2020) and math-word problem (Shi et al., 2022; Chen et al., 2023b) tasks, or they investigate specific capabilities such as machine translation (Zhu et al., 2023), code-switching (Iyer et al., 2023; Zhang et al., 2023) and cross-lingual transfer (Schuster et al., 2019; Baziotis et al., 2023; Tanwar et al., 2023). Finally, despite the efforts of Qin et al. (2023); Huang et al. (2023) and Ranaldi and Zanzotto (2023) in studying the impact of Cross-lingual prompting techniques, they investigated the effect of CoT and derivatives on a limited number of LLMs.

In this paper, we propose a multidimensional approach that unifies these research strands and comprehensively evaluates LLMs' Cross-lingual reasoning abilities. By proposing an analysis of the actual impact of the *reasoning methods*, we aim to investigate the effects of different prompting techniques discerning the characteristics and capabilities of different LLMs in reasoning tasks undertaken in Cross-lingual scenarios.

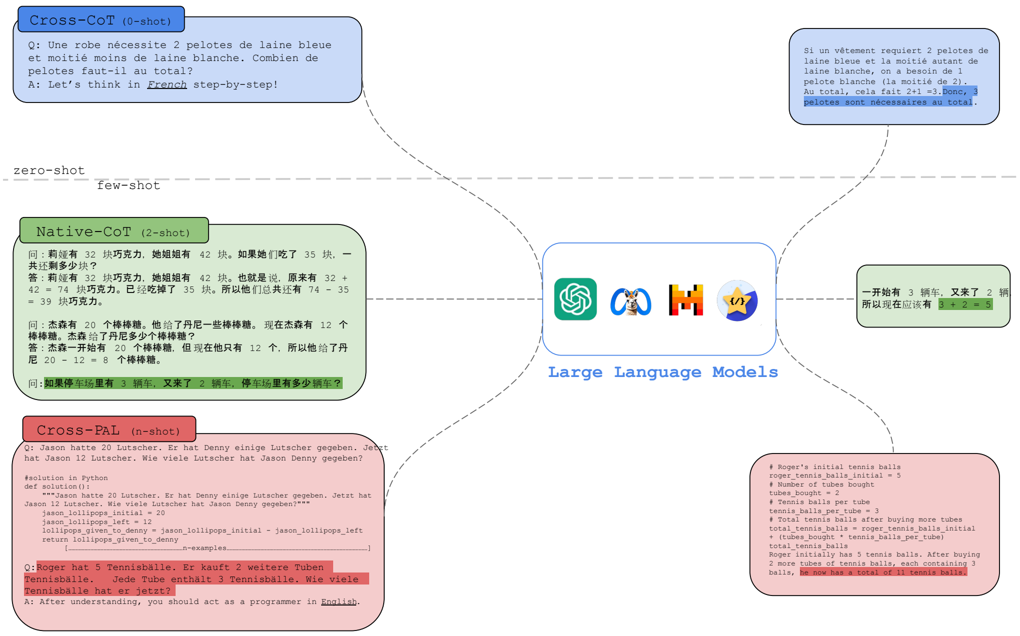


Figure 1: An overview of reasoning methods proposed in our analysis. We study the performances achieved by different LLMs (Section 3.2) on Cross-lingual tasks (Section 3.1) by proposing reasoning methods in zero-shot, and few-shot settings as described in Section 3.3.

This leads to the target research questions, which are the focus of this paper:
RQ1: Does the performance of LLMs in Cross-lingual scenarios benefit from *reasoning methods*?
RQ2: How relevant are the properties of LLMs to empower their performances by applying *reasoning methods*?
RQ3: What is the role of in-context demonstrations?

We answer these questions by proposing a systematic evaluation by prompting different LLMs through *reasoning methods* in Cross-lingual commonsense and math reasoning tasks. In particular, to study the impact of these methods, we propose two aligned versions of CoT and PAL in zero and few-shot settings (as shown in Figure 1). Concerning PAL, we propose a novel Cross-lingual version; for CoT, we extend the work proposed in (Shi et al., 2022). For the results obtained in reasoning tasks, we employ GPTs (OpenAI, 2023) models, and in additional analysis, we include different versions of Llama2 (Touvron et al., 2023) and Mistral (Jiang et al., 2024). On the other hand, we selected StarCoder (Li et al., 2023) and CodeLlama (Rozière et al., 2024) for the coding capabilities. Finally, to establish a robust evaluation, we use different tasks specifically adapted for Cross-lingual contexts. These tasks are best exemplified by MGSM (Shi et al., 2022) and MSVAMP (Chen et al., 2023b), related to mathematical reasoning problems, and XCOPA (Ponti et al., 2020), PAWS-X (Yang et al., 2019) and XLNI (Conneau et al., 2018) concerning commonsense reasoning and natural language inference.

Our main findings are:

- The effectiveness of *reasoning methods* varies considerably depending on the LLMs, the task, and the language. In general, both CoT and PAL improve baseline performances, extending their effectiveness beyond English. However, the benefits of CoT stably seem to emerge only in models with a higher number of parameters (more than 70 billion).

- On the other side of the coin, in mathematical reasoning tasks, the effect of PAL seems to emerge even in smaller models. It is highlighted in models with high proficiency in coding, although stable relations emerge with in-context demonstrations.

- Finally, from the in-depth analysis, we show the importance of Cross-lingual, in-context demonstrations in all proposed tasks. The models that seem to derive the most significant benefit are those with fewer parameters and do not alter the behavior of the bigger models.

Chain-of-Thought	*Cross* **Chain-of-Thought**
Q: Shawn has five toys. For Christmas, he got two toys each from his mom and dad. How many toys does he have now? **A:** He has 5 toys. He got 2 from mom, so he has 5 + 2 = 7 toys. Then he got 2 more from dad, so in total he has 7 + 2 = 9 toys. The answer is 9. **Q:** <Question$_n$ *(in English)*> **A:** <CoT Demonstration$_n$ *(in English)*>	*....few-examples....* (in case of few-shot prompting)
Q: Olivia has $23. She bought five bagels for $3 each. How much money does she have left? **A:**Let's think step-by-step! (in English)	Given the following question, please act as an expert in multi-lingual understanding in Chinese. **Q:** 服务器机房里有九台电脑。从周一到周四，每天又安装了五台电脑。服务器机房里现在有多少台电脑？ After understanding, you should act as an expert in arithmetic reasoning in English. **A:** Let's resolve the task you understand above step-by-step!

Table 1: The Table on the left represents Chain-of-Thought (CoT) prompting in a few-shot setting where demonstrations of CoT answers are provided as input (Wei et al., 2023) or zero-shot where no demonstrations are provided (Kojima et al., 2023) (see QA pair after the dashed line). On the right, Cross-lingual prompting as proposed in (Shi et al., 2022) where, unlike traditional CoT, the questions and demonstrations are in different languages both with each other and the language of the question.

2 Cross-lingual Reasoning Methods

Reasoning methods are popular prompting strategies that elicit Large Language Models (LLMs) to generate multi-step reasoned answers as introduced in Section 2.1. Although these methods demonstrate their functionality in a variety of tasks, evaluations and further studies are primarily conducted in English, leaving other languages unexplored (Section 2.2). To align the gap between languages, we propose a systematic study of the impact of reasoning methods in Cross-lingual settings as described in detail in Section 2.3.

2.1 Reasoning Methods

These methods, best represented by Chain-of-Thought (CoT) (Wei et al., 2023), and Program-Aided Language Models (PAL) (Gao et al., 2022) are popular prompting strategies that elicit LLMs to solve complex problems by simplifying them and breaking down into a series of sub-problems. They operate in zero-shot (Kojima et al., 2023), few-shot (Wei et al., 2023), self-consistent way (Wang et al., 2023) in case of CoT. Instead, in the case of PAL, they use code interpreter (Zhou et al., 2023). Although these methods showed functionality by obtaining solid performances in math and common sense reasoning tasks, the evaluations are mostly conducted in monolingual (English) settings.

2.2 Reasoning Across Languages

Earlier work studied the impact of different languages on CoT prompting. Shi et al. (2022) tested the effectiveness of CoT translations on zero-shot prompts (see Table 1 and Table 5). Qin et al. (2023) inspired by (Huang et al., 2023) and (Wang et al., 2023), proposed two-step CoT prompting (see Table 6). Finally, Ranaldi and Zanzotto (2023) proposed a self-consistent strategy based on a single-step Tree-of-Thouths prompt (Yao et al., 2023) as described in Appendix B. These studies focused on providing actual performance evaluations of CoT and derived methods on LLMs with a high number of parameters, leaving a gap between smaller models and models built to handle multilingual or code-based problem-solving.

2.3 Aligning Reasoning Methods

Inspired by previous work, we take the next step by proposing a multidimensional assessment to analyze the impact of *reasoning methods* on different LLMs chosen by family, capabilities, and purpose of construction (Section 3.2) by using appropriately crafted tasks presented in Section 3.1. The aim is to align previous findings and study whether introducing different languages limits or improves the functionality of these strategies.

Our contribution through the following key points: (i) construction of Cross-lingual evaluations by proposing an extension of PAL and com-

Program-Aided Language Models (PAL)

```
Q: Roger has 5 tennis balls. He buys
2 more cans of tennis balls. Each can
has 3 tennis balls. How many tennis
balls does he have now?
A: Roger started with 5 tennis balls.
   tennis_balls = 5
   2 cans of 3 tennis balls each is
   bought_balls = 2 * 3 tennis balls.
   The answer is
   answer = tennis_balls + bought_balls
   The answer is 11
------------------------------------
Q: Jason had 20 lollipops. He gave Denny some
lollipops. Now Jason has 12 lollipops. How
many lollipops did Jason give to Denny?
A:
```

Cross **Program-Aided Language Models**

.... *few-examples* .... as in (Gao et al., 2022)

```
------------------------------------
Given the following question, please
act as an expert in multi-lingual
understanding in German.
Q: Michael hat 58 Golfbälle. Am Dienstag
hat er 23 Golfbälle verloren. Am Mittwoch
hat er 2 weitere verloren. Wie viele
Golfbälle hat er Mittwoch am Ende des
Tages?
After understanding, you should act as a
programmer in English.
A:
```

Table 2: The Table on the left represents Program-Aided Language Models (PAL) method in a few-shot setting where demonstrations of PAL answers are provided as input (Gao et al., 2022). On the right, our Cross-lingual PAL (Cross-PAL) prompting where, unlike traditional PAL, the questions and demonstrations are in different languages both with each other and the language of the question.

pleting the Cross-lingual CoT (Shi et al., 2022; Huang et al., 2023) methods on different models employing appropriately crafted benchmarking tasks; (ii) fine-grained investigation of the results as far as concern the characteristics of selected LLMs; (iii) analysis of the impact of the presence of high-resources languages in the construction of evaluations.

3 Experimental setup

3.1 Data

In order to study the impact of reasoning methods in Cross-lingual tasks, we use MGSM (Shi et al., 2022), MSVAMP (Chen et al., 2023b), XNLI (Conneau et al., 2018), and PAWS-X (Yang et al., 2019), XCOPA (Ponti et al., 2020).

Understanding tasks To assess Cross-lingual comprehension abilities, we use XNLI and PAWS-X. The first is an extension of Stanford Natural Language Inference (SNLI) (Bowman et al., 2015) across 15 languages and is based on one premise and one hypothesis and requires the model to determine whether the hypothesis is entailed, contradicted, or neutral conditioned on the premise in 15 different languages. The second, Paraphrase Adversaries from Word Scrambling (PAWS-X) (Yang et al., 2019), contains two sentences and requires the model to judge whether they paraphrase each other in seven languages.

Commonsense Reasoning task The Cross-lingual Choice of Plausible Alternatives (XCOPA) (Ponti et al., 2020) is based on one premise and two choices. It asks the model to choose which one is the result or cause of the premise. It covers 11 languages from 11 diverse families.

Arithmetic Reasoning task To evaluate the problem-solving abilities in Cross-lingual scenarios, we use the extension of GSM8K Cobbe et al. (2021) and SVAMP (Patel et al., 2021). Respectively, Multilingual Grade School Math (MGSM) (Shi et al., 2022) and Multilingual Simple Variations on Arithmetic Math word Problems (MSVAMP) (Chen et al., 2023b). In both original cases, the authors proposed a benchmark of mathematical problems in English. The examples have the following structure: a math word problem in natural language and a target answer in Arabic numbers. Shi et al. (2022); Chen et al. (2023b), in their contribution, selected a subset of instances from the official list of examples and translated them manually into 11 different languages, maintaining the structure of the input and output.

Evaluated Languages In our experiments, in order to promote open-source sharing, we use a list of tasks available in different languages; we provide detailed descriptions in Appendix A.

3.2 Models

We evaluate the impact of reasoning methods on different LLMs. Following the previous work, we use three models from the GPT family; moreover, in additional experiments, we introduce other models from the Llama2 and Mistral families and StarCoder2. Hence, complementing previous evaluations, we choose models for (i) multilingual performances achieved by the GPTs (GPT-3.5 and GPT-4) and Llama2 (Llama2-70, -13 and -7) (Ahuja et al., 2023),(ii) the monolingual abilities in mathematical reasoning achieved by Mixtral (Jiang et al., 2024) on GSM8K, and finally, (iii) the proficiency in coding for StarCoder2 (Li et al., 2023), CodeLlama (Rozière et al., 2024), and GPTinstruct (also for results in PAL (Gao et al., 2022; Ye et al., 2023)). The GPT family models were used via API, while the other models were used, the versions available open-source on huggingface. Appendix F describes the parameters and versions used in detail.

3.3 Prompting Methods

In order to conduct the study on robust models and have a term of comparison with the previous results, we start from state-of-art promoting methods and extend the experimental setting by introducing a Cross-lingual version of Program-Aided Language Models (PAL) (Gao et al., 2022) as in Figure 1.

Arithmetic Reasoning Prompts We define two types of prompts for the MGSM and MSVAMP tasks by adapting the CoT and PAL reasoning methods for cross-lingual scenarios. Hence, we adapt CoT to cross-lingual input as proposed in (Qin et al., 2023; Huang et al., 2023) (see Table 1). Concerning PAL, we include cross-lingual demonstrations as shown in Table 2. Moreover, in the additional experimented prompts in Appendices B and E, we evaluated further approaches based on specific language prompting (Shi et al., 2022).

Understanding & Commonsense Prompts While we employ the workflow proposed in previous works for arithmetic tasks by performing experiments with zero and few-shot settings, for understanding and commonsense tasks, we define input templates that lead to the comprehension of LLMs and consequently aid generation. As described in detail in Appendix D, we construct prompts following (Ahuja et al., 2023) using the CoT prompting method to elicit multi-step generations.

Evaluation We evaluate performance using the accuracy score, following the approaches used in (Shi et al., 2022; Gao et al., 2022). In particular, we measure the exact match between generated outputs and labels[1] (Ahuja et al., 2023). The top-p parameter is set to 1 in all processes. We select the Prompting temperature [0, 1] (in Appendix F are reported detailed info).

4 Main Results

Large Language Models (LLMs) benefit from reasoning methods not merely in monolingual contexts (as amply demonstrated in English) but also in multi and Cross-lingual scenarios. These conclusions can mainly be observed from the results obtained on mathematical reasoning, understanding, and commonsense tasks (Table 3). As discussed in Section 4.1, LLMs prompted with both Chain-of-Thought (CoT) and Program-Aided Language Models (PAL) aligned for Cross-lingual tasks generally outperform the baseline prompting approaches.

Although reasoning methods have been shown to be effective in Cross-lingual settings, their operations appear to extend to only some models. As discussed in Section 4.2, more improvements are achieved in the LLMs with higher parameters than in the lower ones. On the other hand, input rationales consistently influence the performance of models with fewer parameters (Section 4.3).

Finally, in Section 4.4, we examine the impact of the presence of high-resource languages on the final performance by discerning the factors that influence the generation of the final response and highlighting the matter of Cross-lingual demonstrations in low-resource settings.

4.1 Reasoning Methods operate across languages

Reasoning methods empower the LLMs' performances in mathematical reasoning tasks, both in multi-lingual and Cross-lingual scenarios. In Table 3, the CoT method (i.e., `"En-CoT"`) consistently outperforms the baselines (i.e., `"Direct"`) aligned results achieved by `PAL` and `En-PAL`. Furthermore, using a cross-lingual generative switch, the best results are displayed in the `Cross-CoT` and `Cross-PAL` methods.

In particular, in results achieved by GPT-4 do not

[1] We extract target labels from the generated answers using regular expressions before calculating the exact match.

Model	Method	Mathematical		Understanding		Commonsense
		MGSM	MSVAMP	XNLI	PAWS-X	XCOPA
GPT-4	`Direct`	67.1	69.2	75.4	68.1	89.0
	`En-CoT`	68.4	71.6	76.1	70.7	91.7
	`En-PAL`	73.4	72.5	-	-	-
	`Cross-CoT`	76.9	73.4	**76.8**	**73.6**	**96.2**
	`Cross-PAL`	**77.0**	**74.3**	-	-	-
GPT-3.5-based	`Direct`	48.5	59.3	62.1	66.4	80.2
	`En-CoT`	57.8	64.3	63.2	67.2	85.3
	`En-PAL`	63.3	63.9	-	-	-
	`Cross-CoT`	65.0	**69.4**	**64.0**	**70.1**	**84.3**
	`Cross-PAL`	**69.3**	68.5	-	-	-
Llama-70-based	`Direct`	45.9	54.0	48.2	58.3	70.2
	`En-CoT`	54.0	56.7	**49.8**	**60.6**	**75.1**
	`En-PAL`	48.8	55.8	-	-	-
	`Cross-CoT`	**57.2**	**60.1**	47.8	57.8	72.5
	`Cross-PAL`	53.1	59.6	-	-	-

Table 3: The average accuracy scores achieved by models proposed in Section 3.2 using *reasoning methods* introduced in Section 3.3 (in **bold** the best performance per model and task). For `GPT-3.5-based`, we reported results achieved by `gpt-3.5-turbo` and `gpt-instruct`, and For `Llama-70-based`, we reported results achieved by `Llama2-70` and `CodeLlama-70`. In Appendices G and H are reported detailed results.

emerge a significant distinction between `En-CoT`, `En-PAL` and the baseline `Direct`, since this model has already shown robust results in many languages (OpenAI, 2023). In contrast, GPT-based and Llama-70-based models appear to obtain noticeable benefits from `En-CoT` and `En-PAL` prompting (complete results in Appendix G). However, the best performances were obtained by eliciting the LLMs' code-switching skills using the `Cross-CoT` and `Cross-PAL` promting method, which significantly increases low-resource language accuracy, discussed in detail Section 4.4.

Although these LLMs benefit the most from introducing reasoning methods in the prompting stage, further improvements are observable even in LLMs with fewer parameters. The complete tables in Appendix H show that Llama2-7, CodeLlama-7, and StarCoder2 outperform the baselines when reasoning methods are used.

These results demonstrate the sensitivity of in-context prompting also in understanding (in Tables 18 and 19) and commonsense task in Table 20. However, some models do not to be benefit-stable as in the case of Mixtral8x7 in Appendix G and Mistral-7 in Appendix H. Hence, in Section 4.2, we investigate this phenomenon in depth and analyze if it is relevant to other tasks as well.

4.2 The Limits of Reasoning Methods

The effective operation of reasoning methods is limited to certain models. In fact, as shown in the complete results in Appendix G for larger LLMs and Appendix H for smaller LLMs, the improvements in using reasoning methods are evident in larger models and less apparent in smaller models. Hence, a distinction between methods and models emerges. The in-context rationales (see `En-CoT` and `En-PAL`) positively affect final performance in mathematical reasoning tasks, with the exception of the Mistral family models as can be seen from Tables 14 and 16. The same phenomenon occurs in XCOPA, PAWS-X, and XNLI (respectively in Table 20 and Figures 18,19), where the same phenomenon can be seen when using the reasoning method `En-CoT`.

Although reasoning methods have brought more benefits than misleading results, these do not extend into cross-lingual prompting, where the smaller LLMs did not achieve the same increment as the larger LLMs and, in some cases, underperformed the baselines (see Mistral-7 and Llama2-7 in Table 16). Finally, a difference in operation between specialized and non-specialized models emerges. Specifically, the specialized models for code (GPT-instruct, CodeLlama, and StarCoder)

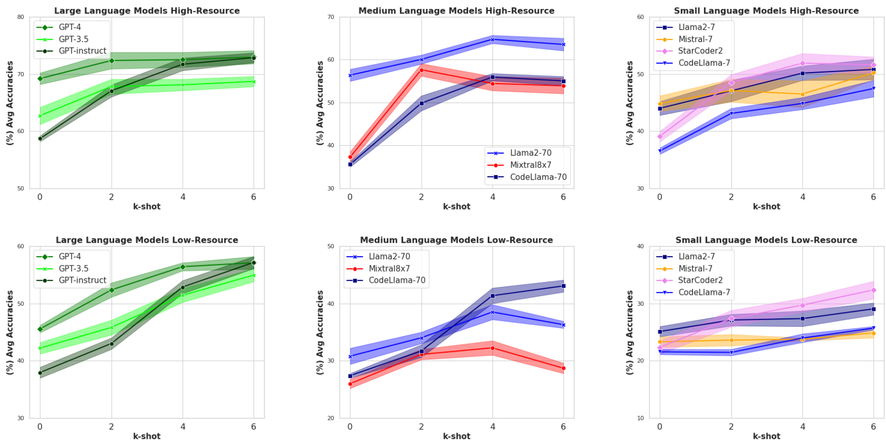


Figure 2: Average accuracies on mathematical reasoning task (MGSM) using methods proposed in (Section 3.3) setting providing in input k-shot demonstrations with k equal to {0,2,4,6}. In Appendix K and Appendix M, detailed results are reported.

seem to exploit in-context demonstrations better when these are closer to their nature, as in the case of PAL, En-PAL, and Cross-PAL.

The improvements in the presence of in-context demonstrations have been extensively studied previously both in English (Chen et al., 2023a) and in mono-lingual and cross-lingual contexts (Yang et al., 2023; Ahuja et al., 2023). However, the role of different type of in-context demonstration (e.g., CoT rationales or PAL) has never been thoroughly analyzed across the languages. Hence, in Section 4.3, we investigate the effect of different types of cross-lingual demonstrations in detail.

4.3 The Role of Demonstrations

In-context demonstrations play a key role in complex scenarios such as multi- and cross-lingual settings because they promote multi-step reasoning across languages. In particular, we investigated the performance trend as in-context demonstrations increased by discerning between the quality and quantity of demonstrations operated. Hence, we repeated the previous experiments focusing on a mathematical task (MGSM), starting with zero- and increasing to 6-shots, as adopted in earlier experiments. The results display that the positive impact of in-context demonstrations across the languages is related to both the quality and quantity of demonstrations used.

Quantitative Impacts The amount of in-context demonstrations seems to be relevant. However, a distinction emerges between models and the number of demonstrations that are de facto useful. In fact, GPT-based models with 4-shots achieve results comparable to 6-shots (see the average accuracies for high-resource languages in Figure 2). This balance does not occur in Llama2-70, CodeLlama-70, and Mixtral, which underperform as in-context demonstrations increase (especially for low-resource languages). Finally, the smaller models (Llama2-7, Mistral-7, CodeLlama-7 StarCoder2) have conspicuous improvements as the number of demonstrations increases. However, there are divergences related to the kind of languages as discussed in Section 4.4.

Qualitative Impacts The quality of in-context demonstrations has beneficial effects on downstream performance, even in cross-lingual prompts. Although previous work has shown the positive trend of introducing quality demonstrations in prompts (Kojima et al., 2023), from the results Figure 3 emerges that CoT rationales are much more influential when delivered in cross-lingual settings (see the generations in Tables 23, 24 and 25). Moreover, Cross-CoT and Cross-PAL outperform both the native (Native-CoT) and English (En-CoT) demonstrations stably. The cross-demonstrations, constructed by combining the power of native demonstrations and elicitation in English, achieve significant gains in all languages (as discussed in Section 4.4). As far as commonsense reasoning tasks are concerned, the same scenario does not occur; in fact, demonstrations in English (En-CoT) seem to be more beneficial than Cross-CoT, which in turn is better than direct prompting.

4.4 The Language Matter

Although in-context demonstrations provide tangible benefits (as discussed in Section 4.2) in proposed tasks, a performance gap arises between low- and high-resources languages[2] (Table 11 reports language distributions in (Common Crawl, 2021)). In particular, the GPTs family models in high-resources settings perform stably with an average of four in-context demonstrations while managing to benefit from more in-context demonstrations (up to 6-shots) in low-resources settings (the larger LLMs in Figure 2 and detailed in Figure 3). In contrast, Llama2-70 and Mixtral8x7 underperform when more demonstrations are involved (4-shot outperforms 6-shot) both in low- and high-resource settings. Moreover, the smaller models, i.e., Llama2-7, Mistral-7, StarCoder2, and CodeLlama-7, perform better as dimsotrations increase in both settings (see smaller LLMs in Figure 2). In conclusion, reasoning methods robustly outperform the baselines (`Direct`) and perform better across languages via cross-lingual methods for math tasks and multilingual methods (i.e., `En-CoT`) in the other tasks.

Finally, to observe whether the origin and method of training limit the impact of reasoning methods across languages, we tested other LLMs described in Appendix O. The results obtained in Figure 22 and discussed in Appendix O show that the methods are scalable to other LLMs and the conclusions are comparable to the discussed results. Although these results are promising, they do not help solve the problem of low-resource languages.

5 Related Work

Large Language Models (LLMs) demonstrate in-context learning abilities (Brown et al., 2020; Min et al., 2022b) to guide LLMs to generate desired task responses, marking the advent of the prompting era and surpassing the age of the intermediate steps in algorithmic and structured reasoning (Roy and Roth, 2015; Ling et al., 2017). Early works challenged the efficacy of few-shot techniques for empowering the prompting phase and downstream performances. In particular, Gao et al. (2022) adapted the original idea of Chain-of-Thought (CoT) (Wei et al., 2023) by considering the proficiency of LLMs in producing code they proposed Program-Aided Language Models (PAL) method.

[2] `high (German, Chinese, French, Russian, Spanish, Japanese) and low (Telogu, Bulgarian, Swali, Thai)`

These prompting approaches, called *reasoning methods*, have demonstrated considerable success, but the findings are limited to a single language (i.e., English). Shi et al. (2022) proposed a multilingual evaluation that Qin et al. (2023) extended to cross-lingual by proposing a prompt mechanism to handle requests in any language and generate English CoT. This approach proposed both single-phase (as a single prompt), also adopted by (Huang et al., 2023) and multi-phase that is based on the self-consistency approach (Wang et al., 2023). Although the mechanism achieves state-of-the-art cross-lingual reasoning, the single-phase promting underperforms in low-resource languages, and the multi-phase prompting defined by cascading prompts is supported far away from the zero- or few-shot CoT concept.

In our work, we propose a multidimensional approach (Ruder et al., 2022) to evaluate LLMs' Cross-lingual reasoning abilities We analyze the impact that *reasoning methods* cause on final performance by studying the role of in-context demonstrations in different languages. Hence, the cornerstones can be outlined by the following points: (i) Analysis of the impact of reasoning methods on mathematical, understanding, and commonsense tasks in different languages using several LLMs (selected by features and scope of construction); (ii) Introduction and adaptation of further methods (i.e., PAL assessment on different languages); (iii) Study of the role of in-context demonstrations by discerning between low-resource and high-resource languages.

6 Conclusion

The benefits of reasoning methods for the English language also emerge across the languages. Our analysis shows that LLMs, appropriately elicited, can deliver structured answers in different languages. Indeed, by adapting two methods, CoT and PAL, to settings in various languages, we revealed that in-context demonstrations play a strategic role by improving performance in direct proportion to their quality without depending on quantity. Our research highlights the need for a customized approach to employing reasoning methods for LLMs in different languages. It supports the demand for a reasonable combination of model scale, reasoning technique, and strategic use of in-context demonstrations to elicit the prospect of LLMs in different language landscapes.

Limitations

Due to the limitations imposed by the evaluation benchmarks and the cost of the OpenAI API, we conducted tests on five tasks and 16 different languages in total, which only scratches the surface of the world's vast array of languages. In addition, our approaches are based on a single-stage prompting approach in English. It should be evaluated Self-consistency prompts (Wang et al., 2023) and using different configurations of cross-lingual in-context demonstrations. Finally, we tested the effectiveness of our method on GPT-based models (closed-source) and several models (open-source). In the future, it will be appropriate to study the generality of our model compared to other closed-source Large Language Models.

Finally, although we have considered and analyzed 9 different families distributed over 22 models in our work, we would like to take a closer look at the performance achieved by language-specific pre-trained models (better known as language-centered). However, at the moment, there are not many open resources comparable in size to those we have analyzed. In the future, we hope these models can be readily available to investigate this phenomenon better.

Ethics Statemets

In our work, ethical topics were not addressed. The data comes from open-source benchmarks, and statistics on language differences in commonly used pre-training data were obtained from official sources without touching on gender, sex, or race differences.

References

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. 2023. MEGA: Multilingual evaluation of generative AI. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267, Singapore. Association for Computational Linguistics.

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai.

Christos Baziotis, Biao Zhang, Alexandra Birch, and Barry Haddow. 2023. When does monolingual data help multilingual translation: The role of domain and model scale.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

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. 2020. Language models are few-shot learners.

Jiuhai Chen, Lichang Chen, Chen Zhu, and Tianyi Zhou. 2023a. How many demonstrations do you need for in-context learning? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11149–11159, Singapore. Association for Computational Linguistics.

Nuo Chen, Zinan Zheng, Ning Wu, Ming Gong, Yangqiu Song, Dongmei Zhang, and Jia Li. 2023b. Breaking language barriers in multilingual mathematical reasoning: Insights and observations.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems.

Common Crawl. 2021. Common crawl 2021. Web. Accessed: 2023-12-12.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2022. Pal: Program-aided language models. *arXiv preprint arXiv:2211.10435*.

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Wayne Xin Zhao, Ting Song, Yan Xia, and Furu Wei. 2023. Not all languages are created equal in llms: Improving multilingual capability by cross-lingual-thought prompting.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2.

Vivek Iyer, Edoardo Barba, Alexandra Birch, Jeff Z. Pan, and Roberto Navigli. 2023. Code-switching with word senses for pretraining in neural machine translation.

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, Lélio 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. 2024. Mixtral of experts.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2023. Starcoder: may the source be with you!

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022a. Rethinking the role of demonstrations: What makes in-context learning work?

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022b. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4.

OpenAI. 2023. Gpt-4 technical report.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.

Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. 2023. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2695–2709, Singapore. Association for Computational Linguistics.

Leonardo Ranaldi and Fabio Massimo Zanzotto. 2023. Empowering multi-step reasoning across languages via tree-of-thoughts.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In *Proceedings of the 2015*

Conference on Empirical Methods in Natural Language Processing, pages 1743–1752, Lisbon, Portugal. Association for Computational Linguistics.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2024. Code llama: Open foundation models for code.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2022. Square one bias in NLP: Towards a multi-dimensional exploration of the research manifold. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2340–2354, Dublin, Ireland. Association for Computational Linguistics.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3795–3805, Minneapolis, Minnesota. Association for Computational Linguistics.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2022. Language models are multilingual chain-of-thought reasoners.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. 2023. Prompting gpt-3 to be reliable.

Eshaan Tanwar, Subhabrata Dutta, Manish Borthakur, and Tanmoy Chakraborty. 2023. Multilingual LLMs are better cross-lingual in-context learners with alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6292–6307, Toronto, Canada. Association for Computational Linguistics.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason

Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Zhao Yang, Yuanzhe Zhang, Dianbo Sui, Cao Liu, Jun Zhao, and Kang Liu. 2023. Representative demonstration selection for in-context learning with two-stage determinantal point process. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5443–5456, Singapore. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. 2023. Satlm: Satisfiability-aided language models using declarative prompting.

Ruochen Zhang, Samuel Cahyawijaya, Jan Christian Blaise Cruz, Genta Indra Winata, and Alham Fikri Aji. 2023. Multilingual large language models are not (yet) code-switchers.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning.

James Zhao, Yuxi Xie, Kenji Kawaguchi, Junxian He, and Michael Xie. 2023. Automatic model selection with large language models for reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 758–783, Singapore. Association for Computational Linguistics.

Aojun Zhou, Ke Wang, Zimu Lu, Weikang Shi, Sichun Luo, Zipeng Qin, Shaoqing Lu, Anya Jia, Linqi Song, Mingjie Zhan, and Hongsheng Li. 2023. Solving challenging math word problems using gpt-4 code interpreter with code-based self-verification.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Multilingual machine translation with large language models: Empirical results and analysis.

A Proposed Task

Dataset	Task	Languages	#Languages
MGSM	mathematical reasoning	`Bengali (bn), Chinese (zh), French (fr), Thai (th) German (de), Japanese (jp), Russian (ru), Telugu (te) Spanish (es), Swahili (sw)`	10
MSVAMP	mathematical reasoning	`Bengali (be), Chinese (zh), French (fr), Thai (th) German (de), Japanese (jp), Russian (ru) Spanish (es), Swahili (sw)`	9
XNLI	natural language inference	`English (en), German (de), Russian (ru), French (fr), Spanish (es), Chinese (zh), Vietnamese (vi), Turkish (tr), Arabic (ar), Greek (el), Thai (th), Bulgarian (bg), Urdu (ur), Swahili (sw), Hindi (hi)`	15
XCOPA	commonsense reasoning	`Chinese (zh), Italian (it), Vietnamese (vi), Indonesian (in), Turkish (tr), Thai (th), Estonian (et), Tamil (ta), Swahili (sw), Haitian (ht), Quechua (qu)`	11
PAWS-X	paraphrase identification	`English (en), German (de), Japanese (jp), French (fr), Spanish (es), Chinese (zh), Korean (ko)`	7

Table 4: Languages present in datasets used in this work.

K K-shot per Model

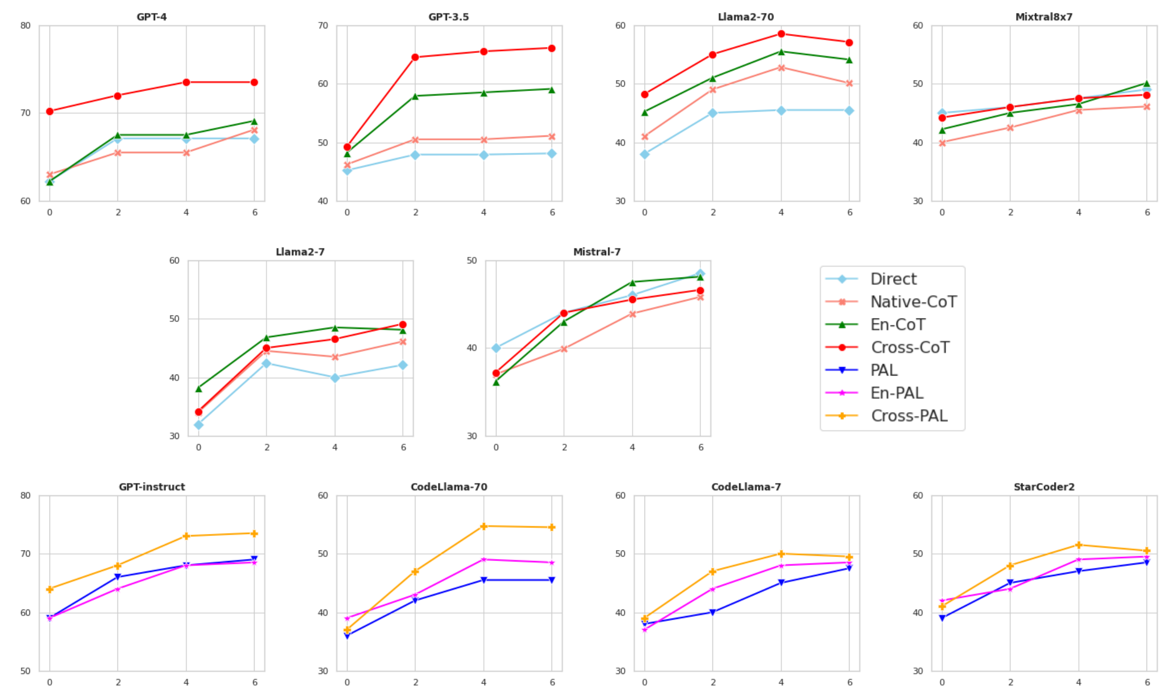


Figure 3: Acciracies (%) on MGSM using methods proposed in (Section 3.3) setting providing in input `k-shot` demonstrations with `k` equal to `{0,2,4,6}`. In contrast to the experiments reported in Appendix M, we here show the performance of each model introduced in Section 3.2

M K-shot per Language using Cross-CoT

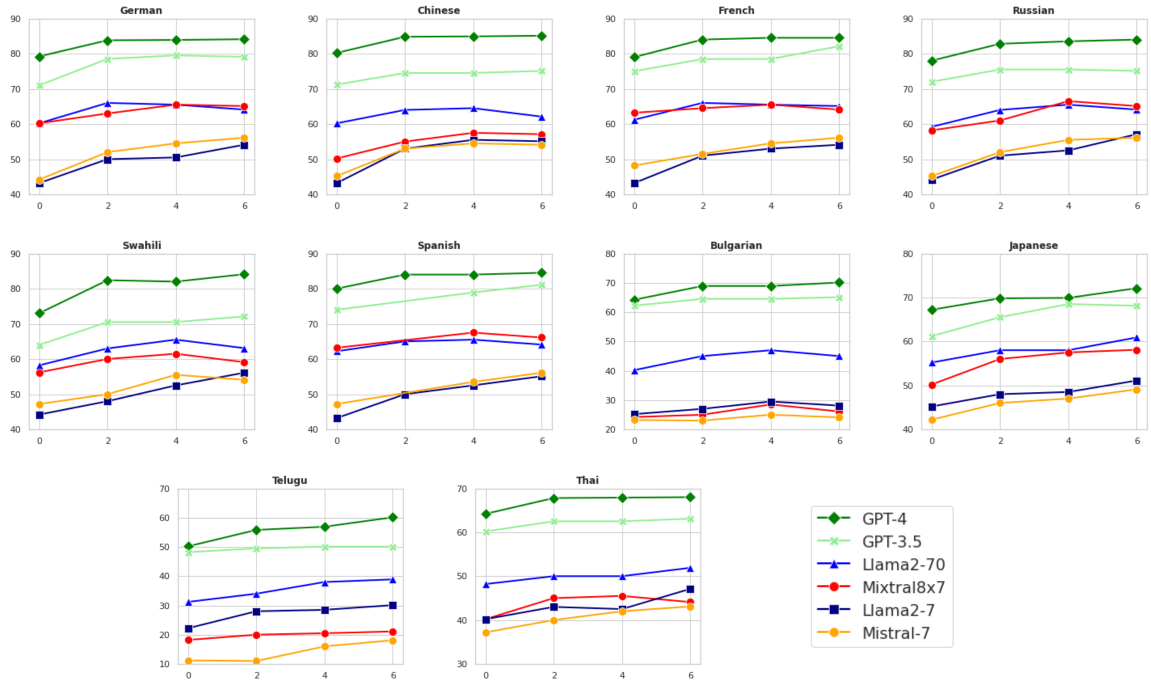


Figure 4: Acciracies (%) on MGSM using `Cross-CoT` (Section 3.3) setting providing in input `k-shot` demonstrations with k equal to {`0,2,4,6`}.

N K-shot per Language using Cross-PAL

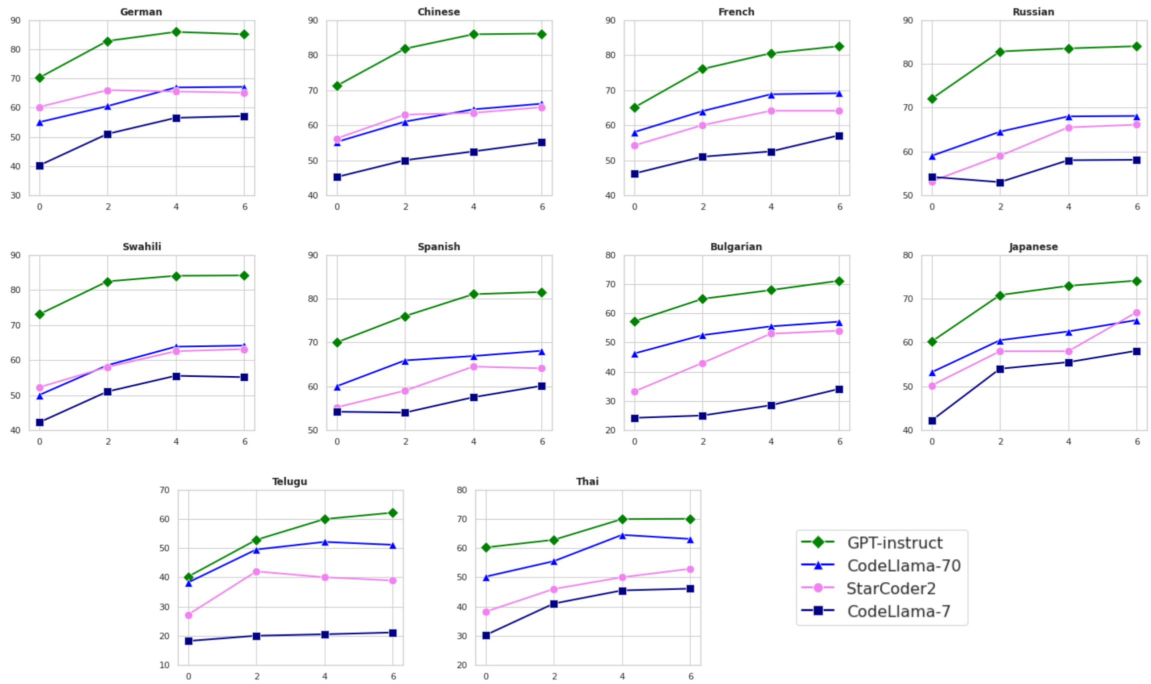


Figure 5: Acciracies (%) on MGSM using `Cross-PAL` (Section 3.3) setting providing in input `k-shot` demonstrations with k equal to {`0,2,4,6`}.

B State-of-art Prompting Methods

`Direct` (Question in Chinese without CoT)

Q: ：罗杰有5 个网球。他又买了2 罐网球。每罐有3 个网球。他现在有多少个网球?
A: 11
Q: 利亚有32 块巧克力，她妹妹有42 块。如果她们吃了35 块，她们一共还剩下多少块?
A:

`Native-CoT` (Question and CoT Answer in Chinese)

Q: 罗杰有5 个网球。他又买了2 罐网球。每罐有3 个网球。他现在有多少个网球?
A: 罗杰一开始有5 个球。2 罐各3 个网球就是6 个网球。5 + 6 = 11。答案是11。
Q: 利亚有32 块巧克力，她妹妹有42 块。如果她们吃了35 块，她们一共还剩下多少块?
A: 让我们一步步思考

`En-CoT` (Question in Chinese and CoT Answer in English)

Q: 罗杰有5 个网球。他又买了2 罐网球。每罐有3 个网球。他现在有多少个网球?
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.
Q: 利亚有32 块巧克力，她妹妹有42 块。如果她们吃了35 块，她们一共还剩下多少块?
A: `Let's think step-by-step!`

Table 5: Chain-of-Thought as proposed in (Shi et al., 2022) (for simplicity we have reduced the shot but the original is 6-shot). Given a problem in specific language, the following prompts are `Direct`, `Native-CoT` (without additional languages) and `En-CoT`, the original question in specific language with answers in English.

CLIP First-Step

Please act as an expert in multi-lingual understanding in [Specific Language L_s].
Question: [Given sentence X in L_s]
Let's understand the task in [Target Language L_t] step-by-step!

CLIP Second-Step

After understanding, you should act as an expert in mathematics in [Language L_t].
Let's resolve the task you understand above step-by-step!

Table 6: CLIP (Qin et al., 2023) where the prompt is split into two phases: there is the alignment of the different languages, and then, there is the solving mechanism for the specific language.

Cross-ToT

Simulate the collaboration of $\{n\}$ mathematicians answering a question in their mother tongue: L_1, L_2, ... and L_n. They all start Step1 from a separate thought process, step by step, each explaining their thought process. Following Step1, each expert refines and develops their thought process by comparing themselves with others. This process continues until a definitive answer to the question is obtained.
Question: [Question in Language L_1]
Answer: [num].

Table 7: `Cross-ToT` prompting (Ranaldi and Zanzotto, 2023) that using Tree-of-Thoughts method elicit the model to produce multi-step reasoning processes in different languages.

C Prompting Methods Arithmetic Reasoning Tasks

In this work, as introduced in Section 3, we propose the Cross-lingual extension of Program-Aided Language Models (Cross-PAL) as shown in Table 2 (detailed in Appendix E), and a Cross-lingual version of CoT as shown in Table 1. In detail, in both settings, the prompt is a few-shots as proposed in (Wei et al., 2023) for CoT and in (Gao et al., 2022) for PAL, respectively; however, unlike the previous versions, the question-answer pairs (the answers are a CoT demonstration) are proposed in the languages evaluated in each task. Moreover, we use additional configurations as proposed by Shi et al. (2022): `"Direct"` prompt, i.e., question and answer in the original language; the `"Native-CoT"` prompt, i.e., question and answer CoT in the original language; the `"En-CoT"` prompt specific language question and answer CoT in English (see prompts in Appendix B). Furthermore, in order to analyse the effect of reducing the in-context examples down to zero-shots we propose additional settings esemplifing the number and the typology of demonstrations that compose the prompt.

D Prompts for Understanting & Commonsense Reasoning Tasks

As far as prompts for natural language understanding and commonsense reasoning tasks are concerned, we follow the methods proposed by state-of-the-art works. Hence, following Ahuja et al. (2023), to construct prompts that lead Large Language Models (LLMs) to produce stable and structured answers, we define a sequence consisting of *Task Instruction*, *Demonstration*, and *Task Problem*. In particular, the *Task Instruction* is the initial instruction that defines the type of task and the desired answer. Then, there is a body composed of Demonstrations that are related to the number of shots. For example, in the few-shot settings such as CoT proposed in (Shi et al., 2022), the demonstrations are composed of questions and desired outputs. Finally, the final part consists of questions about the tasks we are analyzing. As in Appendix B, we propose `Direct`, `En-CoT` and `Cross-CoT` configurations while we do not use PAL as it is not suitable for this type of task. In Table 12, we report the selected templates. Table 23, 24 and 25 report the demonstrations, input and outputs generated.

E Program-Aided Language Models Prompts

In this paper, as introduced in Section 3.3, we propose a novel Cross-lingual extension of the Program-Aided Language Models (Gao et al., 2022) (Cross-PAL) method. The following tables show the prompts used for the final evaluation.

Program-Aided Language Models (PAL)

```
Q: Roger has 5 tennis balls. He buys 2 more
cans of tennis balls. Each can has 3 tennis
balls. How many tennis balls does he have now?
A: Roger started with 5 tennis balls.
   tennis_balls = 5
   2 cans of 3 tennis balls each is
   bought_balls = 2 * 3 tennis balls.
   The answer is
   answer = tennis_balls + bought_balls
   The answer is 11
Q: Kyle bought last year's best-selling book
for $19.50. This is with a 25% discount from the
original price. What was the original price?
A:
```

Table 8: This is an example prompt of the PAL method proposed by (Gao et al., 2022).

En-PAL

```
Q: 罗杰有5 个网球。他又买了2 罐网球。每罐有3 个
网球。他现在有多少个网球?
A: Roger started with 5 tennis balls.
   tennis_balls = 5
   2 cans of 3 tennis balls each is
   bought_balls = 2 * 3 tennis balls.
   The answer is
   answer = tennis_balls + bought_balls
   The answer is 11
Q: 克莱尔每天早上用3 颗鸡蛋做一份煎蛋饼当早
餐。四周内她能吃多少打鸡蛋?
A:
```

Table 9: In En-PAL we use the same setting proposed in Table 5 but in contrast to En-PAL we use PAL demonstrations.

Cross **Program-Aided Language Models**

```
Q: Michael hat 58 Golfbälle. Am Dienstag hat
er 23 Golfbälle verloren. Am Mittwoch hat er
2 weitere verloren. Wie viele Golfbälle hat er
Mittwoch am Ende des Tages?
A: Michael hat 58 Golfbälle.
   initial = 58
   Am Dienstag verlor er 23 Golfbälle
   lost_tuesday = 23
   Am Mittwoch verlor er 2 Golfbälle
   lost_wednesday = 2
   Golfbälle abzüglich der verlorenen
    answer = initial - lost_tuesday
     - lost_wednesday
    Die Antwort ist 33
```

Table 10: In Cross-PAL, we use the same setting proposed in Table 9 but in contrast to En-PAL, we use PAL demonstrations in the same language of the question.

F Model and Hyperparameters

In our experimental setting, as introduced in Section 3.2, we propose different LLMs: (i) three models from the GPT family (OpenAI, 2023): GPT-3.5 (`gpt-3.5-turbo-0125`), Codex (`gpt-3.5-turbo-instruct`) and GPT-4 (`gpt-4`); (ii) four models from the Llama-2 family (Touvron et al., 2023): Llama2-7b, Llama2-70b, CodeLlama-7 and CodeLlama-70; (iii) two models of the MistralAI family: Mistral-7b and Mixtral (Jiang et al., 2024); (iv) finally, StarCoder2-15b (Li et al., 2023). In particular, GPTs models are used via API, while for the others, we used versions of the quantized to 4-bit models that use GPTQ (see detailed versions in Table 21)

Furthermore, we have added additional LLMs in the additional experiments presented in the Apendix O. These models are two from Orca2 (Mukherjee et al., 2023), two from Yi (AI et al., 2024), two models of the Google (Team et al., 2024), three from Wizard (Luo et al., 2023), and three from Tulu (Ivison et al., 2023) families.

As discussed in the limitations, our choices are related to reproducibility and the cost associated with non-open-source models. We use closed-source API and the 4-bit GPTQ quantized version of the model on four 48GB NVIDIA RTXA600 GPUs for all experiments performed only in inference.

Finally, the generation temperature used varies from $\tau = 0$ of GPT models to $\tau = 0.5$ of Llama2s. We choose these temperatures for (mostly) deterministic outputs, with a maximum token length of 256. The other parameters are left unchanged as recommended by the official resources. We will release the code and the dataset upon acceptance of the paper.

Language	Percentage
English (en)	46.3%
Russian (ru)	6.0%
German (de)	5.4%
Chinese (zh)	5.3%
French (fr)	4.4%
Japanese (ja)	4.3%
Spanish (es)	4.2%
Other	23.1%

Table 11: Language distribution of CommonCrawl (Common Crawl, 2021).

Task Instruction for:

XNLI
You are an NLP assistant whose purpose is to solve Natural Language Inference (NLI) problems in {**evaluated language**}. NLI is the task of determining the inference relation between two (short, ordered) texts: entailment, contradiction, or neutral. Answer as concisely as possible in the same format as the examples below:
XCOPA
You are an AI assistant whose purpose is to perform open-domain commonsense causal reasoning in {**evaluated language**}. You will be provided a premise and two alternatives, where the task is to select the alternative that more plausibly has a causal relation with the premise. Answer as concisely as possible in the same format as the examples below:
PWAS-X
You are an NLP assistant whose purpose is to perform Paraphrase Identification in {**evaluated language**}.The goal of Paraphrase Identification is to determine whether a pair of sentences have the same meaning. Answer as concisely as possible in the same format as the examples below:

Table 12: *Task Instruction for* XNLI, XCOPA and PAWS-X as proposed in (Ahuja et al., 2023). List of the Basic Prompt is in Table 13

Benchmark	**#Test**	**Final Prompt**
MGSM	250	Q: {problem}
MSVAMP	1000	Q: {problem}
XCOPA	200	Here is a premise: {premise}. What is the {question}? Help me pick the more plausible option: -choice1: {choice1}, -choice2: {choice2}
XNLI	200	{premise}. Based on the previous passage, is it true that {hypothesis}? Yes, No, or Maybe?
PAWS-X	200	Sentence 1: {sentence1} Sentence 2: {sentence2} Question: Does Sentence 1 paraphrase Sentence 2? Yes or No?

Table 13: The prompt of each task (excepted for MGSM and MSVAMP) that is systematically added following the instructions we defined in Table 12. The column **#Test** denotes the number of instances for each language in the test set proposed by the authors. The constructions of these tasks are derived from translations (manual or automatic) of subsets of the original monolingual versions (in English) as explained in Section 3.1.

G Results Arithmetic Reasoning Tasks Larger Models

The following evaluations were made by prompting the models presented in Section 3.2 with the methods presented in Section 3.3 (see Appendix B and Appendix E where the prompts are described in detail).

Model	Method	de	zh	fr	ru	sw	es	bn	ja	te	th	Avg
	Direct	56.0	60.0	62.0	62.0	48.0	61.2	33.6	52.8	7.6	42.2	48.5
	Native-CoT	70.0	59.6	64.4	62.4	50.0	61.2	26.4	54.4	22.0	46.6	51.6
	En-CoT	73.6	63.2	70.0	65.6	55.2	69.6	50.4	60.6	40.0	48.0	59.8
GPT-3.5	Cross-CoT	79.2	74.2	**78.0**	**75.8**	**68.2**	**73.6**	**65.2**	67.6	49.2	62.8	68.8
	PAL	62.8	52.6	58.2	58.2	48.6	60.2	28.8	58.6	34.4	50.0	51.3
	En-PAL	68.4	59.6	64.6	59.8	51.2	66.4	35.6	61.8	38.2	55.8	56.1
	Cross-PAL	**80.8**	**78.2**	77.8	68.8	65.2	70.8	63.6	**69.8**	53.0	**64.4**	**69.2**
	PAL	67.2	58.0	62.0	62.4	51.6	63.6	32.0	61.6	39.4	54.8	55.3
$\text{GPT}_{\text{instruct}}$	En-PAL	72.2	62.8	66.8	62.0	53.4	64.6	31.8	60.0	40.4	59.6	57.6
	Cross-PAL	**82.0**	**78.8**	76.6	**70.2**	66.4	72.6	63.4	**69.8**	**55.0**	64.0	**70.0**
	Direct	52.2	55.0	58.2	60.0	46.4	58.6	30.2	48.6	9.2	41.0	45.9
Llama2-70	Native-CoT	63.8	61.0	60.2	58.2	51.4	65.0	28.8	50.6	28.4	44.2	51.0
	En-CoT	64.0	61.4	61.6	61.4	50.6	63.8	35.0	54.2	37.4	49.0	54.0
	Cross-CoT	**65.2**	**62.8**	**64.8**	**64.6**	**53.8**	**68.2**	**44.0**	56.4	**38.2**	**51.2**	**57.2**
	PAL	53.0	57.2	52.8	52.4	48.2	53.8	24.0	50.0	12.8	46.8	45.0
CodeLlama-70	En-PAL	57.6	57.6	57.2	56.8	48.3	54.8	27.8	53.6	26.0	46.4	48.8
	Cross-PAL	62.8	60.4	59.2	58.6	48.2	54.6	28.2	**56.6**	34.6	48.6	53.1
	Direct	**58.2**	**62.4**	**64.4**	**62.8**	**54.2**	**62.8**	**35.0**	54.2	12.8	44.6	51.2
Mixtral8x7	Native-CoT	56.8	58.2	57.6	56.8	50.2	62.0	25.8	55.6	18.6	45.4	48.7
	En-CoT	55.8	59.4	58.6	58.4	51.0	63.0	44.8	56.8	**22.2**	**46.6**	**51.6**
	Cross-CoT	57.6	56.8	58.2	57.2	53.0	61.2	28.4	**58.6**	20.0	45.2	49.5

Table 14: Accuracies (%) on MGSM using the reasoning methods described in Appendix C (for each model, we reported best performances per language and per method in **bold**).

Model	Method	de	zh	fr	ru	sw	es	bn	ja	th	Avg
	Direct	60.3	66.2	63.5	60.3	59.2	69.2	9.6	68.9	36.2	59.3
	Native-CoT	68.9	76.5	77.8	68.5	66.3	74.5	12.1	73.1	43.5	62.4
	En-CoT	73.9	78.4	78.2	70.9	68.4	74.6	14.4	74.0	46.1	64.3
GPT-3.5	Cross-CoT	78.4	**78.6**	79.3	74.8	70.4	75.2	**41.0**	**76.2**	51.4	69.4
	PAL	64.3	72.5	73.0	67.3	64.2	72.8	7.8	67.2	42.4	59.3
	En-PAL	66.7	74.3	74.8	70.3	67.2	73.1	19.3	71.7	47.2	62.9
	Cross-PAL	82.3	76.9	80.2	75.7	71.6	76.8	37.7	74.5	50.2	69.5
	PAL	65.3	73.1	73.7	68.6	66.1	71.9	8.7	68.6	41.2	59.7
$\text{GPT}_{\text{instruct}}$	En-PAL	67.8	74.7	75.3	71.0	67.6	74.2	20.0	72.1	48.3	63.4
	Cross-PAL	**82.6**	78.2	**81.6**	**76.8**	**73.1**	**77.2**	40.3	76.1	**53.4**	**70.2**
	Direct	55.9	65.2	64.6	59.8	58.3	68.6	8.5	67.5	37.8	54.0
Llama2-70	Native-CoT	60.7	64.8	60.9	60.5	59.1	67.3	13.2	66.8	36.7	54.8
	En-CoT	63.5	66.3	62.8	61.7	60.2	66.0	20.3	65.9	40.3	56.7
	Cross-CoT	66.5	**69.5**	**65.9**	**64.6**	62.5	**68.6**	**30.7**	**69.3**	44.1	**60.1**
	PAL	61.3	61.2	59.3	56.8	57.8	63.7	14.6	64.2	38.1	53.3
CodeLlama-70	En-PAL	67.4	64.2	60.3	62.1	60.2	63.9	20.6	63.4	40.2	55.8
	Cross-PAL	**70.4**	67.6	65.3	64.3	**63.2**	65.2	23.4	65.8	**45.8**	59.6
	Direct	63.5	**67.5**	64.2	59.7	60.1	68.3	15.1	68.5	38.2	56.2
Mixtral8x7	Native-CoT	63.1	66.7	65.3	60.2	61.4	**69.5**	15.6	69.3	**40.1**	56.8
	En-CoT	**66.2**	67.3	**66.8**	61.7	62.5	68.9	**16.2**	70.0	40.1	**57.6**
	Cross-CoT	64.8	64.7	65.4	**62.3**	**62.8**	66.2	15.6	**70.3**	37.1	56.5

Table 15: Accuracies (%) on MSVAMP using the reasoning methods described in Appendix C(for each model, we reported best performances per language and per method in **bold**).

H Results Arithmetic Reasoning Tasks Smaller Models

Model	Method	de	zh	fr	ru	sw	es	bn	ja	te	th	Avg
							MGSM					
Llama2-7	Direct	48.4	50.2	54.0	56.8	42.0	54.8	28.0	46.2	5.4	38.4	42.5
	Native-CoT	54.8	51.0	55.4	57.6	48.8	58.4	27.4	49.2	20.0	41.6	46.1
	En-CoT	56.0	55.2	56.4	60.2	**51.0**	60.2	30.0	**50.2**	22.6	43.8	48.0
	Cross-CoT	**53.8**	54.4	**56.2**	**57.6**	50.4	**62.6**	27.4	50.0	28.8	45.2	**48.9**
CodeLlama-7	PAL	48.6	54.4	49.2	49.6	45.8	50.2	20.4	48.6	6.4	40.0	41.0
	En-PAL	50.8	56.0	51.0	50.2	49.4	51.4	24.6	49.8	16.2	44.0	43.4
	Cross-PAL	52.4	54.8	55.0	57.2	49.0	61.8	**30.2**	49.8	26.6	**45.6**	48.5
Mistral-7	Direct	**56.0**	**60.6**	**62.0**	**60.2**	52.0	**60.0**	**34.4**	**52.0**	12.0	47.4	**49.5**
	Native-CoT	54.2	58.4	60.2	58.6	51.4	58.6	32.6	50.2	12.2	47.8	48.0
	En-CoT	55.6	59.2	61.4	59.0	**52.2**	58.8	32.4	51.0	**14.0**	**48.0**	48.4
	Cross-CoT	54.2	57.4	60.0	58.4	50.2	58.6	32.0	51.8	12.4	47.8	48.2
StarCoder2	PAL	54.6	56.8	52.4	52.6	48.8	54.0	24.6	48.6	14.0	46.8	45.1
	En-PAL	**56.2**	**58.4**	54.0	54.8	**50.2**	56.4	26.2	52.8	16.2	48.0	47.3
	Cross-PAL	54.2	55.2	**54.6**	57.0	50.0	**60.2**	**28.0**	**50.2**	**25.0**	**49.0**	**48.4**
							MSVAMP					
Llama2-7	Direct	51.2	57.3	57.1	51.0	50.9	56.3	10.4	60.2	-	30.1	46.8
	Native-CoT	52.8	58.7	58.2	52.3	51.7	57.0	11.7	62.8	-	32.3	48.6
	En-CoT	55.6	**59.8**	**60.0**	52.6	**54.2**	56.9	18.8	**63.7**	-	**34.5**	**51.0**
	Cross-CoT	53.4	57.7	58.0	51.6	51.3	57.2	**19.7**	63.4	-	32.3	48.8
CodeLlama-7	PAL	54.2	58.4	56.3	53.4	52.3	57.2	9.3	58.5	-	31.3	47.4
	En-PAL	55.6	59.3	57.4	54.2	53.1	58.0	10.7	58.9	-	32.4	48.6
	Cross-PAL	**55.8**	59.4	57.3	**55.4**	54.0	**58.8**	17.5	57.6	-	29.2	49.6
Mistral-7	Direct	**52.6**	58.7	**59.0**	52.3	51.4	55.9	8.8	62.1	-	**32.7**	48.2
	Native-CoT	50.7	57.2	56.8	52.0	52.1	56.8	9.1	**63.7**	-	31.8	47.8
	En-CoT	51.3	**58.6**	57.2	**53.2**	**52.8**	**57.6**	**10.4**	62.1	-	32.3	**48.6**
	Cross-CoT	50.8	57.3	57.6	53.0	52.4	54.3	6.7	59.3	-	28.6	46.2
StarCoder2	PAL	54.5	59.3	57.4	54.3	52.6	58.7	11.6	58.3	-	32.0	48.6
	En-PAL	56.2	**60.2**	58.2	**55.4**	**53.2**	**59.0**	14.5	59.2	-	32.7	49.6
	Cross-PAL	**57.0**	59.6	**58.5**	56.3	51.3	57.4	**15.1**	**58.9**	-	**34.2**	**50.2**

Table 16: Accuracies (%) on MGSM and SVAMP of further models using the reasoning methods described in Appendix C (in **bold** the best performance of each model).

I Results Arithmetic Reasoning Tasks GPT-4

Model	Method	de	zh	fr	ru	sw	es	bn	ja	te	th	Avg
							MGSM					
GPT-4	Direct	78.0	79.2	83.0	78.4	76.2	82.2	38.8	72.0	18.4	65.4	67.1
	Native-CoT	78.8	79.6	84.2	79.2	77.1	83.4	34.0	76.2	25.4	66.2	68.4
	En-CoT	80.6	80.0	84.4	81.2	78.2	84.2	56.0	78.4	45.6	68.6	73.7
	Cross-CoT	83.0	83.2	85.2	83.4	80.0	83.2	**60.6**	80.6	**57.0**	68.2	76.9
	PAL	79.0	80.2	82.6	77.8	78.2	82.8	41.0	77.2	41.2	64.0	70.1
	En-PAL	80.2	81.4	83.0	79.6	79.2	83.2	55.0	79.2	51.0	67.4	73.4
	Cross-PAL	**84.4**	**83.6**	**85.0**	**83.8**	**81.6**	**85.0**	58.8	**81.2**	56.2	**70.2**	**77.0**
							MSVAMP					
GPT-4	Direct	74.1	73.6	81.2	76.3	70.5	77.2	36.0	70.5	-	65.9	69.2
	Native-CoT	74.6	74.2	81.8	76.2	71.4	78.1	38.0	71.2	-	66.3	70.2
	En-CoT	76.7	76.3	82.6	77.8	71.2	81.3	39.6	71.8	-	67.2	71.6
	Cross-CoT	81.3	77.5	83.4	78.2	73.1	82.1	42.8	73.6	-	68.5	73.4
	PAL	75.8	76.9	83.2	78.0	72.4	79.6	40.2	72.0	-	66.3	71.7
	En-PAL	77.9	78.8	83.2	78.1	72.1	82.4	38.2	72.5	-	69.4	72.5
	Cross-PAL	**82.4**	**78.6**	**83.7**	**78.5**	**73.7**	**82.7**	**43.2**	**74.5**	-	**70.2**	**74.3**

Table 17: Accuracies (%) on MGSM and SVAMP of GPT-4 on first 100 questions for each language using the reasoning methods described in Appendix C.

J Performances on XNLI

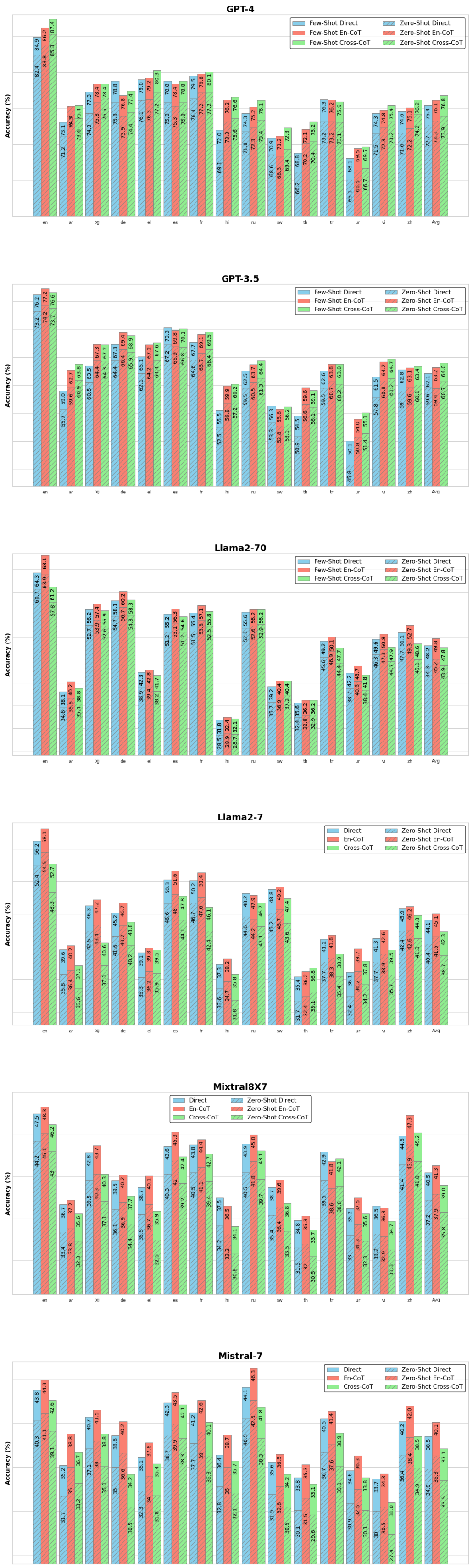


Table 18: Accuracies XNLI using 4-shot and 0-shot prompts.

L Performances on PAWS-X

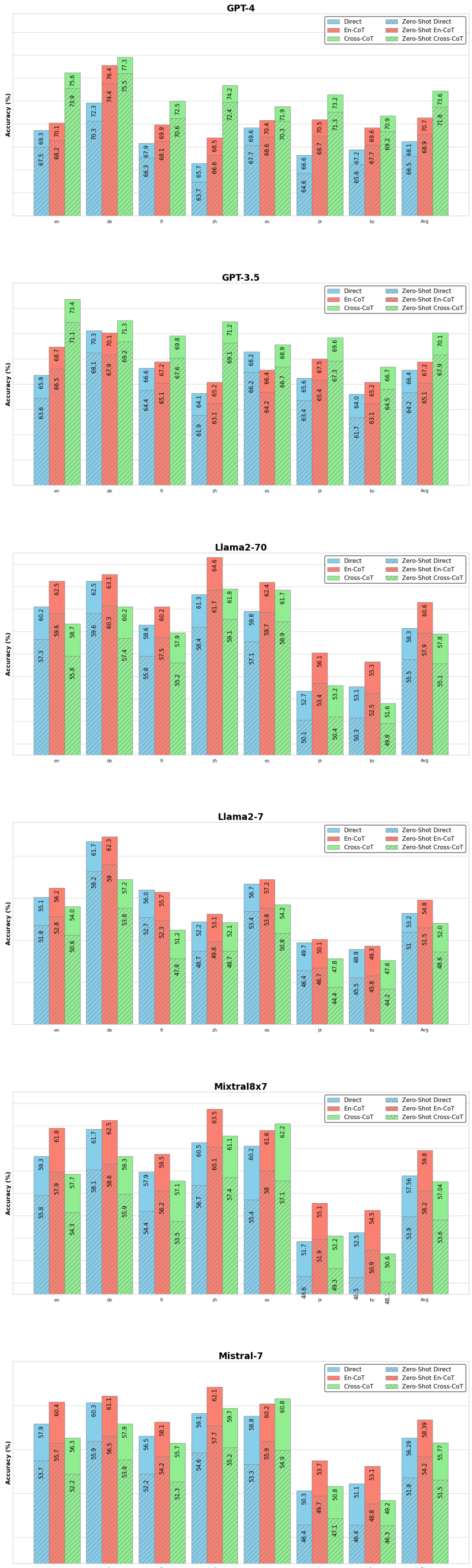


Table 19: Accuracies PAWS-X using 4-shot and 0-shot prompts.

Model	et	ht	id	it	qu	sw	ta	th	tr	vi	zh	Avg
GPT-4												
`Direct`	98.8	93.2	97.6	99.8	58.6	94.4	79.6	87.8	97.4	86.2	92.6	89.0
`En-CoT`	95.8	94.0	96.6	98.2	80.0	95.2	84.6	88.0	93.4	85.2	93.6	91.7
`Cross-CoT`	97.8	95.2	96.6	95.0	84.8	93.8	85.8	91.8	96.6	87.2	94.0	96.2
GPT-3.5												
`Direct`	90.6	72.0	90.4	95.2	54.6	82.0	59.0	77.6	91.0	83.6	90.4	80.2
`En-CoT`	92.4	78.2	91.6	96.8	81.4	81.6	64.8	80.2	93.6	85.2	94.0	85.3
`Cross-CoT`	94.0	79.6	92.2	96.4	82.6	82.0	63.2	82.0	93.8	86.0	93.4	84.4
Mixtral8x7												
`Direct`	82.5	68.0	81.6	54.5	83.1	60.3	78.1	81.9	80.5	74.2	70.6	74.2
`En-CoT`	81.7	66.5	79.3	53.5	82.9	61.3	80.8	82.4	79.8	74.7	70.3	73.9
`Cross-CoT`	80.7	67.1	77.3	54.2	82.0	60.7	80.2	80.3	79.2	73.5	69.2	73.1
Llama2-70												
`Direct`	80.4	66.2	79.8	82.4	52.8	81.6	58.4	76.0	79.2	73.0	69.2	70.2
`En-CoT`	84.2	68.8	80.4	84.6	55.2	82.8	60.6	78.4	80.4	74.4	71.6	75.1
`Cross-CoT`	79.8	66.0	78.2	81.6	51.2	80.2	57.8	77.2	80.4	73.6	70.6	72.7
Llama2-7												
`Direct`	39.6	32.5	58.4	55.8	47.2	34.6	47.4	33.2	43.0	59.6	50.4	45.4
`En-CoT`	42.8	36.6	60.2	56.2	50.0	36.8	48.6	34.8	44.2	60.8	51.6	47.1
`Cross-CoT`	40.8	36.2	57.8	56.2	48.4	33.0	47.0	34.4	44.2	60.2	51.6	46.2
Mistral-7												
`Direct`	42.6	36.5	60.1	57.8	48.7	37.3	49.2	36.6	45.2	59.3	51.2	47.7
`En-CoT`	41.9	37.1	59.8	57.2	50.1	38.2	49.7	38.5	46.3	60.1	52.3	48.2
`Cross-CoT`	39.7	36.5	57.6	56.8	49.6	38.4	48.7	37.5	45.2	59.4	50.7	47.3
HUMAN (Ponti et al., 2020)	98.2	96.4	100.0	97.0	94.8	99.0	98.6	98.2	96.4	98.4	96.6	97.6

Table 20: Accuracies (%) on XCOPA (Ponti et al., 2020) using the reasoning methods described in Appendix C. (`Direct`, `En-CoT` and `Cross-CoT` as introduced in Section 3.3.

O Scalability to further LLMs

In this experiment, we study the performance of additional Large Language Models (LLMs) that were not considered in the principal analysis. The following models were chosen for performances obtained in mathematical reasoning tasks (in the case of WizardMath (Luo et al., 2023)), for performances obtained in specific languages other than English (in the case of Tulu (Ivison et al., 2023) and Yi (AI et al., 2024)), and finally for abilities also shown in functions with a limited number of parameters (in the case of gemma (Team et al., 2024)). In order to make the results comparable with the previous ones, we used the same experimental setup introduced in Section 3. However, we only produced evaluations for a few instances of the MSGM task (we used the same instances as those used for GPT-4). This further experiment observes whether the selected models perform comparably to those discussed in Section 4.
From Figure 22, it can be observed that just as it happened in previous LLMs (results detailed in Table 16), the following points emerge:

- Reasoning methods operate between languages. As discussed in Section 4.1, almost all LLMs prompted via `En-CoT` stably overperform the baselines, i.e., `Direct`.
- However, there are limitations, as yet discussed in Section 4.2. Models with fewer parameters (see Orca and Gemma) underperform when the quality of in-context prompts is more articulated (`Direct` vs. `En-CoT` case).
- While the smaller models appear not to benefit under varying in-context demonstrations, the larger models (in these experiments, they are average LLMs not comparable to GPT-4) outperform when the `Cross-CoT` promting strategy is used, as happens to the results discussed in Section 4.

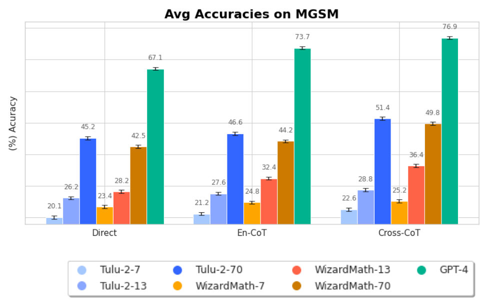
In conclusion, these additional experiments have shown that this approach is scalable to other LLMs and that, generally, the findings obtained by testing the models presented in Section 3.2 are scalable to further models.

P Models Vesions

Model	Version
Llama-2-7-chat Llama-2-13-chat Llama-2-70-chat	meta-llama/Llama-2-7b meta-llama/Llama-2-13b meta-llama/Llama-2-70b
gemma-2 gemma-7	google/gemma-2b google/gemma-7b
Orca-2-7 Orca-2-13	microsoft/Orca-2-7b microsoft/Orca-2-13b
Mistral-7-instruct Mixtral	mistralai/Mistral-7B-Instruct-v0.2 TheBloke/Mixtral-8x7B-Instruct-v0.1-GPTQ
Yi-6b Yi-34b	TheBloke/Yi-6B-GPTQ TheBloke/Yi-6B-GPTQ
Tulu-2-7 Tulu-2-13 Tulu-2-70	TheBloke/tulu-2-7B-GPTQ TheBloke/tulu-2-13B-GPTQ TheBloke/tulu-2-70B-GPTQ
WizardMath-7 WizardMath-13 WizardMath-70	TheBloke/WizardMath-7B-V1.0-GPTQ TheBloke/WizardMath-13B-V1.0-GPTQ TheBloke/WizardMath-70B-V1.0-GPTQ
StarCoder2 CodeLlama-70 (7)	bigcode/starcoder2-15b TheBloke/CodeLlama-70B (7)-Instruct-GPTQ
GPT-3.5-turbo GPT-instruct GPT-4	OpenAI API OpenAI API OpenAI API

Table 21: List the versions of the models proposed in this work, which can be found on huggingface.co. We used all the default configurations proposed in the repositories for each model.

Q Performances other LLMs



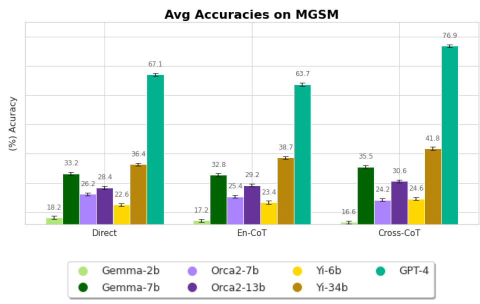


Table 22: Average accuracies across specific languages of further LLMs by using prompting pipelines proposed in Section 3.3.

En-CoT

Q: 罗杰有5 个网球。他又买了2 罐网球。每罐有3 个网球。他现在有多少个网球?
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.
Q: 问题：服务器机房里有九台电脑。从周一到周四，每天又安装了五台电脑。服务器机房里现在有多少台电脑?
A: There are 4 days from monday to thursday. 5 computers were added each day. That means in total 4 * 5 = 20 computers were added. There were 9 computers in the beginning, so now there are 9 + 20 = 29 computers. The answer is 29.
Q: 问题：肖恩有五个玩具。圣诞节他从他爸爸妈妈那里各得到了两个玩具。他现在有多少个玩具?
A: He has 5 toys. He got 2 from mom, so after that he has 5 + 2 = 7 toys. Then he got 2 more from dad, so in total he has 7 + 2 = 9 toys. The answer is 9.
Q: 问题：利亚有32 块巧克力，她妹妹有42 块。如果她们吃了35 块，她们一共还剩下多少块?
A: Leah had 32 chocolates and Leah's sister had 42. That means there were originally 32 + 42 = 74 chocolates. 35 have been eaten. So in total they still have 74 - 35 = 39 chocolates. The answer is 39.
Q: 问题：如果停车场里有3 辆车，又来了2 辆车，停车场里有多少辆车?
A: There are 3 cars in the beginning, 2 more arrive, so now there should be 3 + 2 = 5 cars. The answer is 5.
Q: 问题：杰森有20 根棒棒糖。他给了丹尼一些棒棒糖。现在杰森有12 根棒棒糖。杰森给了丹尼多少根棒棒糖?
A: Jason started with 20 lollipops, but now he only has 12, so he gave Denny 20 - 12 = 8 lollipops. The answer is 8.
Q: 制作一件袍子需要2 匹蓝色纤维布料和这个数量一半的白色纤维布料。它一共需要用掉多少匹布料?
A:

GPT-4

制作一件袍子需要2匹蓝色纤维布料和这个数量一半的白色纤维布料，也就是2匹的一半即1匹白色纤维布料。因此，总共需要的布料数量为2匹蓝色纤维布料加上1匹白色纤维布料，即：
2(蓝色)+1(白色)=3(总共) 答案是，
制作一件袍子总共需要3匹布料。

GPT-3.5

制作一件袍子需要2匹蓝色纤维布料和这个数量一半的白色纤维布料。这意味着需要用掉2匹蓝色和1匹白色布料。所以一共需要用掉2 + 1 = 3匹布料。

Mixtral8x7

Making a robe requires 2 pieces of blue fabric and half as much white fabric as blue fabric. So you need 2 (blue fabric) + 2/2 (white fabric) = 2 (blue fabric) + 1 (white fabric) = 3 pieces of fabric in total. The answer is 3.

Llama2-7

[:..] 制作一件袍子需要 two π blue cotton fabrics and this amount half white cotton fabric. How much fabric do they need in total? A: To make one shirt, you need 2 pi (π) blue cotton fabrics and half of them is equal to 1 pi (π) white cotton fabric. So, in total, you need 2 * π + (1/2) * π = 3π white cotton fabrics.

Mistral-7

To make one suit, it requires 2 pieces of blue fabric and half as many pieces of white fabric. Therefore, it takes 2 (blue) + 2 (half of what 2 is for white) = 3.5. Since we cannot have a half piece of fabric, let's assume it needs an additional 0.5 pieces to cover this requirement. So, it takes a total of 3.5 x 2 = 7 pieces of fabric to create one suit. The answer is 7.

Table 23: Example output answers generated by different LLMs prompted via En-CoT (Section 3.3). In general, LLMs produce good answers while Smaller LLMs (for example see Llama2-7) produce misleading answers.

.

`Cross-CoT`

Q: 罗杰有5 个网球。他又买了2 罐网球。每罐有3 个网球。他现在有多少个网球?
`A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.`
Q: 问题：服务器机房里有九台电脑。从周一到周四，每天又安装了五台电脑。服务器机房里现在有多少台电脑?
`A: There are 4 days from monday to thursday. 5 computers were added each day. That means in total 4 * 5 = 20 computers were added. There were 9 computers in the beginning, so now there are 9 + 20 = 29 computers. The answer is 29.`
Q: 问题：肖恩有五个玩具。圣诞节他从他爸爸妈妈那里各得到了两个玩具。他现在有多少个玩具?
`A: He has 5 toys. He got 2 from mom, so after that he has 5 + 2 = 7 toys. Then he got 2 more from dad, so in total he has 7 + 2 = 9 toys. The answer is 9.`
Q: 问题：利亚有32 块巧克力，她妹妹有42 块。如果她们吃了35 块，她们一共还剩下多少块?
`A: Leah had 32 chocolates and Leah's sister had 42. That means there were originally 32 + 42 = 74 chocolates. 35 have been eaten. So in total they still have 74 - 35 = 39 chocolates. The answer is 39.`
Q: 问题：如果停车场里有3 辆车，又来了2 辆车，停车场里有多少辆车?
`A: There are 3 cars in the beginning, 2 more arrive, so now there should be 3 + 2 = 5 cars. The answer is 5.`
Q: 问题：杰森有20 根棒棒糖。他给了丹尼一些棒棒糖。现在杰森有12 根棒棒糖。杰森给了丹尼多少根棒棒糖?
`A: Jason started with 20 lollipops, but now he only has 12, so he gave Denny 20 - 12 = 8 lollipops. The answer is 8.`

Q: 制作一件袍子需要2 匹蓝色纤维布料和这个数量一半的白色纤维布料。它一共需要用掉多少匹布料? After understanding, you should act as an expert in arithmetic reasoning in English.
`A: Let's resolve the task you understand above step-by-step!`

`GPT-4`

To calculate the total amount of fabric required to make one robe based on the given conditions: [...] To find the total amount of fabric needed for one robe, you sum the requirements for blue and white fabric: Total fabric = Blue fabric + White fabric = 2 bolts (blue) + 1 bolt (white) = 3 bolts So, **3 bolts** of fabric are needed in total to make one robe: 2 bolts of blue fabric and 1 bolt of white fabric.

`GPT-3.5`

To determine how many pieces of fabric are needed in total, we first calculate the amount of white fabric required. Since the blue fabric needed is 2 pieces, the white fabric needed would be half of that, which is 1 piece. [...] If you want to know how many pieces of fabric are needed for a certain number of robes, you would multiply the number of robes by the total pieces of fabric needed for each robe. For example, if you want to make 5 robes, you would need $5 \times 3 = 15$ pieces of fabric in total.

`Mixtral8x7`

A robe requires 2 pieces of blue fabric and half the amount of white fabric as compared to the blue fabric. First, determine the amount of white fabric required. Since the white fabric is half the amount of blue fabric, and there are 2 pieces of blue fabric, the robe needs 0.5 * 2 = 1 piece of white fabric. Next, add up the amount of both types of fabric to find the total amount required. The robe needs 2 pieces of blue fabric and 1 piece of white fabric, so it needs 2 + 1 = 3 pieces of fabric in total. Therefore, a robe requires 3 pieces of fabric in total (2 pieces of blue fabric and 1 piece of white fabric).

`Llama2-7`

We know that the total amount of fabric required is 20 pieces (since the problem states that there are 20 pieces of fabric in total). So, we can set up the equation: 2x = 20 Now, let's divide both sides of the equation by 2 to get: x = 10 So, the blue fabric requires 10 pieces, and the white fabric requires 10 / 2 = 5 pieces. In total, there are 10 + 5 = 15 pieces of fabric required to make one shirt.

`Mistral-7`

To make one suit, you need 2 pieces of blue fabric and half as many pieces of white fabric. So you need 2 (blue) + 2 (half of the number of white pieces) = 2 + 1 = 3 pieces of blue fabric and 2 (half of the number of white pieces) = 2 pieces of white fabric in total. Therefore, you need 3 (blue) + 2 (white) = 5 pieces of fabric in total. The answer is 5.

Table 24: Example output answers generated by different LLMs proposed in our work prompted via `Cross-CoT` (Section 3.3). In contrast to the generation in Table 23 Llama-2-7 does not deliver misleading answer.

.

Cross-PAL

Q: Jason hatte 20 Lutscher. Er hat Denny einige Lutscher gegeben. Jetzt hat Jason 12 Lutscher. Wie viele Lutscher hat Jason Denny gegeben?
A: #solution in Python

```
def solution():
    jason_lollipops_left = 12
    jason_lollipops_initial = 20
    lollipops_given_to_denny = jason_lollipops_initial - jason_lollipops_left
    return lollipops_given_to_denny
```

The answer is 8.
(...more in-context demonstrations...)

Q: Kylar geht ins Kaufhaus, um Gläser für seine neue Wohnung zu erwerben. Ein Glas kostet 5 US-Dollar, aber jedes weitere Glas kostet nur 60% des Ausgangspreises. Kylar möchte 16 Gläser kaufen. Wie viel muss er dafür ausgeben?
A: After understanding, you should act as a programmer in English.

GPT-4

A: #solution in Python

```
def total_cost_for_glasses():
    first_glass_price = 5
    subsequent_glass_price = 0.60 * first_glass_price
    total_glasses = 16
    cost = first_glass_price + (total_glasses - 1) * subsequent_glass_price
    return cost
```

The answer is 16

GPT-3.5

To calculate the total cost for Kylar's purchase of 16 glasses, we can break it down into two parts: the cost of the first glass and the cost of the additional glasses.
The cost of the first glass is $5.
For each additional glass beyond the first one, the cost is 60% of the original price, which is 0.6 * $5 = $3.
So, for the remaining 15 glasses, the cost would be 15 * $3 = $45.
Adding the cost of the first glass to the cost of the additional glasses, the total cost would be $5 + $45 = $50.
Therefore, Kylar would need to spend $50 to purchase 16 glasses for his new apartment.

CodeLlama-70

A: #solution in Python

```
def cost_of_glasses(glasses):
"""A function to calculate the cost of glases."""
    glasses_cost = 0
    glass_price = 5
    for glass_index in range(glasses):
      glasses_cost += glass_price
      glass_price = 0.6 * glass_price
    return glasses_cost
```

The answer is 16

CodeLlama-7

A: #solution in Python

```
def glass_cost(n): return 5 * n + 0.6 * 5 * (n - 1) price = glass_cost(16) print("Kylar will spend around $154.00 to buy 16 glasses.")
```

Table 25: Example output answers generated by different LLMs prompted via Cross-PAL (Section 3.3). Generations are structured and guided by program languages.