The Impact of Language on Arithmetic Proficiency: A Multilingual Investigation with Cross-Agent Checking Computation

Chung-Chi Chen,[1] Hiroya Takamura,[1] Ichiro Kobayashi,[2] Yusuke Miyao[3]
[1] Artificial Intelligence Research Center, AIST, Japan
[2] Ochanomizu University, Japan
[3] University of Tokyo, Japan
c.c.chen@acm.org, takamura.hiroya@aist.go.jp,
koba@is.ocha.ac.jp, yusuke@is.s.u-tokyo.ac.jp

Abstract

This paper critically examines the arithmetic capabilities of Large Language Models (LLMs), uncovering significant limitations in their performance. Our research reveals a notable decline in accuracy for complex calculations involving large numbers, with addition and subtraction tasks showing varying degrees of proficiency. Additionally, we challenge the notion that arithmetic is language-independent, finding up to a 10% difference in performance across twenty languages. The study also compares self-verification methods with cross-agent collaborations, showing that a single model often outperforms collaborative approaches in basic arithmetic tasks. These findings suggest a need to reassess the effectiveness of LLMs in tasks requiring numerical accuracy and precision.

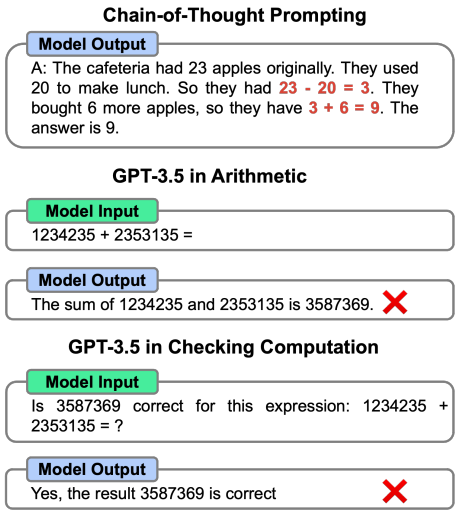

Figure 1: An example of arithmetic in LLM's output in Wei et al. (2022), and an example of the failure case of LLM in arithmetic and checking computation.

1 Introduction

Large language models (LLMs) have garnered significant attention over the past year. Several studies have re-evaluated various tasks to assess the capabilities of general-purpose LLMs (Wadhwa et al., 2023; Zhang et al., 2023; Ho et al., 2023). A topic of particular interest is mathematical and numerical reasoning (Wei et al., 2022; Imani et al., 2023; Gaur and Saunshi, 2023; Davis, 2024). Figure 1 illustrates an instance where LLMs generate step-by-step operational expressions while solving a math word problem, named Chain-of-Thought Prompting (Wei et al., 2022). While previous research indicates improved performance by LLMs in solving math word problems, there is a scarcity of discussion on whether LLMs truly comprehend the operations they generate. This paper delves into this issue through extensive experimentation and reveals a notable limitation of LLMs in arithmetic.

Unlike other semantic tasks such as humor estimation (Hossain et al., 2020) or emotion prediction (Milkowski et al., 2021), where different labels may emerge due to language and cultural variations, arithmetic is typically considered language-free and culture-free, as the same expression should yield a consistent answer regardless of these factors. In this study, we investigate twenty languages and demonstrate that this assumption does not hold in practice. Our findings reveal that the overall performance can vary by up to 10% in accuracy simply by altering the language when utilizing LLMs for arithmetic tasks.

Conversely, addition and subtraction are fundamental yet critical tasks in arithmetic. As depicted in Figure 1, it is commonly assumed in prior research that LLMs are capable of solving such elementary calculations. Contrary to this belief, our study reveals a significant decline in performance for calculations involving more than five digits in addition and more than four digits in subtraction. Furthermore, we observe a 20% discrepancy in accuracy between addition and subtraction tasks. These findings underscore the need to reassess the

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), pages 631–637

extent to which LLMs genuinely comprehend the principles of basic arithmetic.

Finally, checking computation is a crucial step in human arithmetic processing. We initially investigate different prompts to examine the extent to which performance alters with different approaches. Besides self-verification by the same model, our study also delves into cross-agent checking. Contrary to prior research, which indicates that multi-agent communication can enhance performance in contexts such as software development (Qian et al., 2023) and generated-text evaluation (Chan et al., 2023), our findings suggest that a single model surpasses cross-agent collaboration in simple arithmetic tasks. This challenges the prevailing notion that collaborative approaches always yield superior results in NLP tasks.

2 Related Work and Preliminary

Arithmetic computation forms the cornerstone of mathematical capability. Earlier studies (Wies et al., 2023; Liu and Low, 2023) classify arithmetic tasks into two groups: learnable and unlearnable, and Dziri et al. (2024) demonstrated that LLMs fail at multi-digit multiplication. Tasks categorized as learnable include copying, splitting, comparison, ordering, addition, subtraction, and n-digit versus 1-digit multiplication/division. It is anticipated that model performance would be robust when trained specifically on these learnable tasks. Supporting this, Chen et al. (2023a) provides evidence for the comparison task, where models achieve a 99% accuracy rate after straightforward fine-tuning with artificially generated datasets. However, this falls outside the purview of our paper, as our focus is on the capabilities of general-purpose LLMs trained with commonly available resources. In this study, we specifically investigate addition and subtraction within a multilingual context, a subject seldom addressed in previous research.

On the other hand, checking computation is another seldom-explored area of prior studies. Drawing inspiration from Berglund et al. (2023), which demonstrated that LLMs trained on the premise "A is B" struggle to comprehend "B is A" (reversal curse), our research investigates the validity of these findings in arithmetic tasks. Advancing this inquiry, we observe that communicative agents exhibit superior performance compared to the use of a single LLM in various tasks, as noted in many recent studies (Hong et al., 2023; Chen et al., 2023b; Qian et al., 2023; Chan et al., 2023). Building upon this trend, our study delves into the realm of cross-agent checking computation. Our study demonstrates that LLMs currently lack the capability for self-correction in basic arithmetic scenarios, even through LLM interaction.

3 Experimental Setting

3.1 Dataset

In this research, we create an extensive test set comprising 39,708 instances for experimental analysis. Each instance consists of two numbers, ranging from 1 to 16 digits, combined with either an addition or subtraction operator. Examples from the dataset include simple expressions like "1 + 1 = " and more complex ones such as "2468 - 1357 = ". The dataset is evenly split, with 50% of the instances being addition expressions and the remaining 50% subtraction expressions. Instead of presenting equations directly to the LLMs, we employ a standardized prompt: *Answer the following expression, please only reply with the answer: [Expression]*. This prompt is translated and used across 20 different languages: English, Spanish, French, German, Simplified Chinese, Traditional Chinese, Russian, Japanese, Italian, Dutch, Korean, Portuguese, Swedish, Finnish, Danish, Polish, Hindi, Turkish, Greek, and Thai. The input to the model combines both the prompt and the arithmetic expression. This approach allows us to assess the LLMs' arithmetic capabilities in a controlled and consistent manner. We evaluate the performance based on the accuracy.

3.2 Approach

In this study, we primarily utilize GPT-3.5[1] for experimental purposes and compare its performance with PaLM-2[2] using English instances. To assess the impact of language on arithmetic performance, GPT-3.5 is employed to process 39,708 instances across 20 different language settings, amounting to a total of 794,160 instances. Since PaLM-2 is limited to English, a corresponding set of English instances is used for comparative analysis.

Furthermore, we investigate whether LLMs can verify their calculations and whether cross-LLM verification enhances performance. In this experiment, the response from the Answerer (either ChatGPT or PaLM-2) is input into the prompt of the

[1] https://chat.openai.com
[2] https://developers.generativeai.google/

Overall			Addition			Subtraction		
Rank	Language	Acc.	Rank	Language	Acc.	Rank	Language	Acc.
1	**English**	62.44	1	Thai	67.60	1	**English**	60.64
2	Japanese	62.40	2	Korean	66.51	2	Japanese	60.60
3	Trad. Chinese	61.57	3	Turkish	66.38	3	Trad. Chinese	59.76
4	Dutch	61.21	4	German	65.33	4	Dutch	58.42
5	German	61.19	5	Spanish	64.60	5	Russian	57.34
6	Spanish	60.66	6	Portuguese	64.28	6	German	57.06
7	Italian	59.93	7	Danish	64.27	7	Spanish	56.71
8	Russian	59.92	8	**English**	64.24	8	Italian	55.96
9	Portuguese	59.86	9	Japanese	64.21	9	Portuguese	55.45
10	Turkish	59.54	10	Dutch	64.01	10	Finnish	54.17
11	Danish	59.01	11	Italian	63.89	11	Polish	54.17
12	Sim. Chinese	58.47	12	Swedish	63.87	12	Sim. Chinese	54.10
13	Polish	58.35	13	Trad. Chinese	63.38	13	Danish	53.75
14	Swedish	58.16	14	Sim. Chinese	62.83	14	Greek	53.12
15	Finnish	57.94	15	French	62.69	15	Turkish	52.70
16	Thai	57.94	16	Polish	62.54	16	Swedish	52.46
17	Greek	57.81	17	Greek	62.51	17	French	51.11
18	French	56.90	18	Russian	62.49	18	Thai	48.27
19	Korean	56.28	19	Finnish	61.71	19	Korean	46.04
20	Hindi	51.32	20	Hindi	61.27	20	Hindi	41.37
Average		59.05	Average		63.93	Average		54.16
Standard Deviation		2.52	Standard Deviation		1.62	Standard Deviation		4.83

Table 1: GPT-3.5 performance in arithmetic using prompts in different languages (%). Trad. and Sim. Chinese denote traditional and simplified Chinese. Acc. denotes accuracy.

	Overall				Addition				Subtraction			
	All	1-5 digits	6-8 digits	16 digits	All	1-5 digits	6-8 digits	16 digits	All	1-5 digits	6-8 digits	16 digits
GPT-3.5	62.44	93.40	57.06	25.08	64.24	98.26	51.41	33.61	60.64	88.53	62.71	16.54
PaLM-2	81.51	97.88	87.63	31.76	89.91	98.56	96.50	54.01	73.10	97.19	78.76	9.51

Table 2: GPT-3.5 vs. PaLM-2 (%).

Verifier (either ChatGPT or PaLM-2), who is then tasked with verifying the accuracy of the answer. If the response is incorrect, the Verifier is expected to provide the correct solution.

4 Evaluation Results

4.1 Multilingual Examination

Basic arithmetic is universally recognized as a fundamental aspect of common sense, expected to yield consistent results irrespective of geographical or cultural differences. This section posits that arithmetic performance remains relatively stable, regardless of the language employed in the task.

Table 1 offers substantial evidence challenging this assumption. Firstly, arithmetic performance in English surpasses that of other languages, albeit marginally, with respective scores of 62.44%, 64.24%, and 60.64% in overall, addition, and subtraction tasks. Secondly, a significant disparity exists between the highest (English) and lowest (Hindi) performing languages, with a maximum performance gap of 11.22%. Thirdly, GPT-3.5 exhibits superior performance in addition compared to subtraction across all languages, with a higher standard deviation noted in subtraction scores among different languages. Fourthly, there is a notable divergence in the arithmetic abilities of traditional Chinese and simplified Chinese, particularly in subtraction, suggesting limited transferability of arithmetic skills across even closely related languages.

These observations highlight several topics for future exploration. (1) Our findings reveal that the arithmetic capabilities of LLMs hover just above the 60% threshold. This has implications for numerical reasoning studies presuming LLM proficiency in computing expressions, as illustrated in Figure 1; these studies might benefit from focusing on enhancing basic arithmetic skills. (2) The language used significantly affects arithmetic performance, underscoring the need to consider linguistic variables in numeracy assessments and to develop language-independent methods for solving mathematical problems.

4.2 Checking Computation

Computation checking represents a critical capability in arithmetic, with the underlying hypothesis being that LLMs performance can be en-

	Answerer	Verifier	Overall	Addition	Subtraction	Improvement
Self-Checking	GPT-3.5		62.42	64.02	60.82	-0.02
	PaLM-2		73.64	81.18	66.10	-7.87
Cross-Agent Checking	GPT-3.5	PaLM-2	73.25	78.25	68.25	**10.81**
	PaLM-2	GPT-3.5	76.75	88.37	65.13	-4.76

Table 3: Experimental results of checking computation (%). Positive values signify overall performance enhancement, while negative values indicate a decline in performance.

hanced through effective computation checking. This section explores two distinct approaches: self-checking and cross-model checking. Self-checking involves using the same LLM for both computation and verification, while cross-model checking entails employing different LLMs as the answer provider and verifier.

To perform cross-agent checking, we experiment with PaLM-2, which only supports English at this time. According to Table 2, PaLM-2 outperforms GPT-3.5. Further analysis, categorized by the number of digits in the computational tasks, reveals that both LLMs excel with numbers smaller than 10^6. However, GPT-3.5's performance declines with larger numbers. In contrast, PaLM-2 still performs well in addition instances but also drops in subtraction instances. Regarding huge numbers (16 digits), the performances of both LLMs drop significantly.

Table 3 details the results of computation checking. It is observed that LLMs exhibit poorer performance in self-checking scenarios. Notably, when PaLM-2 functions as both the answerer and verifier, there is a significant drop in performance. Additionally, while employing PaLM-2 to verify GPT-3.5's computations yields better outcomes than GPT-3.5 alone, the post-verification performance (73.25%) still falls short of PaLM-2's solo performance (81.51%).

These findings offer insights for arithmetic tasks with recent trends in multi-agent approaches (Qian et al., 2023; Chan et al., 2023). Our results indicate that in simple arithmetic tasks, a single model approach is superior to cross-agent collaboration. Furthermore, these findings highlight the existing challenges in self-checking computations for even high-performing LLMs like PaLM-2, which, despite its robust computational abilities, cannot fully rectify all erroneous instances from GPT-3.5 that are correctly resolved when exclusively employing PaLM-2. Finally, this phenomenon can also be considered a type of reversal curse in arithmetic contexts (Berglund et al., 2023). It potentially affects the efficacy of number-aware fact-conflicting hallucination detection, including the detection of exaggerated information (Chen et al., 2019). Future research focused on number-aware tasks should consider this phenomenon.

	Carry	Non-Carry	Borrow	Non-Borrow
GPT-3.5	63.60	93.63	59.34	84.92
PaLM-2	89.89	91.04	71.99	93.68

Table 4: Performance on basic arithmetic concepts (%).

Model	Input	Overall	Addition	Subtraction
GPT-3.5	Expression Only	51.64%	**64.85%**	38.43%
	English Prompt	**62.44%**	64.24%	**60.64%**
GPT-4	Expression Only	**89.24%**	92.41%	**86.08%**
	English Prompt	86.06%	**92.63%**	79.16%
PaLM-2	Expression Only	79.96%	89.16%	70.75%
	English Prompt	**81.51%**	**89.91%**	**73.10%**
Gemini	Expression Only	75.19%	81.00%	69.38%
	English Prompt	**77.41%**	**85.03%**	**69.79%**

Table 5: Impact of language on arithmetic proficiency.

5 Discussion

5.1 Carry and Borrow

In this section, we categorize the instances into two groups: (1) those requiring a carry (borrow) concept for question resolution, and (2) non-carry (non-borrow) instances. The results are presented in Table 4. Irrespective of the language model used, there is a notable decrease in performance for instances necessitating a carry (borrow) concept. Particularly in scenarios involving the borrow concept, both GPT-3.5 and PaLM exhibit markedly inferior performance compared to non-borrow instances. This observation highlights a deficiency in the generalization capabilities of auto-regressive language models, suggesting that the borrow concept may not be adequately learned during current training processes. Future research should focus on developing tailored approaches to address this limitation in handling arithmetic problems with language models.

5.2 Using Pure Expression

In previous sections, the influence of various languages on numeracy was discussed. This section

Overall			Addition			Subtraction		
Rank	Language	Acc.	Rank	Language	Acc.	Rank	Language	Acc.
1	Russian	87.12%	1	Russian	92.66%	1	Japanese	81.58%
2	Japanese	87.09%	2	**English**	92.63%	2	Russian	81.54%
3	Polish	86.87%	3	Polish	92.55%	3	Polish	81.20%
4	Turkish	86.54%	4	Japanese	92.51%	4	Turkish	80.58%
5	Spanish	86.32%	5	Portuguese	92.50%	5	Spanish	80.13%
6	Trad. Chinese	86.20%	6	Italian	92.45%	6	Trad. Chinese	79.95%
7	**English**	86.06%	7	Spanish	92.45%	7	Greek	79.67%
8	Greek	85.86%	8	Dutch	92.44%	8	Danish	79.28%
9	Danish	85.76%	9	Trad. Chinese	92.36%	9	**English**	79.16%
10	Thai	85.52%	10	Danish	92.28%	10	Thai	78.76%
11	Portuguese	85.22%	11	German	92.25%	11	Hindi	78.19%
12	Italian	85.11%	12	Turkish	92.15%	12	Finnish	78.06%
13	German	85.07%	13	Thai	92.14%	13	German	77.99%
14	Finnish	85.01%	14	Swedish	92.09%	14	Portuguese	77.93%
15	Swedish	84.91%	15	Greek	91.99%	15	Italian	77.84%
16	French	84.61%	16	Finnish	91.80%	16	Swedish	77.43%
17	Dutch	84.55%	17	French	91.71%	17	French	77.39%
18	Korean	82.72%	18	Korean	88.83%	18	Dutch	76.60%
19	Hindi	81.65%	19	Hindi	86.87%	19	Korean	76.43%
20	Sim. Chinese	77.45%	20	Sim. Chinese	84.20%	20	Sim. Chinese	70.71%
Average		84.98%	Average		91.44%	Average		78.52%
Standard Deviation		2.23%	Standard Deviation		2.22%	Standard Deviation		2.40%

Table 6: GPT-4 performance in arithmetic using prompts in different languages (%).

further explores the impact of language on models' numeracy by conducting experiments with purely symbolic expressions to determine if the absence of natural language affects the outcomes. Additionally, two more models, Gemini and GPT-4, were included in the experiment for a more comprehensive discussion.

Table 5 presents the experimental results. Notably, three out of the four models exhibited improved overall performance when arithmetic questions were posed in natural language (English). A closer examination reveals distinctions between two model families (GPT-3.5/GPT-4 and PaLM-2/Gemini). Both PaLM-2 and Gemini showed enhanced performance in addition and subtraction tasks when questions were posed in language. Conversely, GPT-3.5 and GPT-4 demonstrated only marginal differences under various settings. However, for subtraction tasks, natural language significantly enhanced GPT-3.5's performance while detrimentally affecting GPT-4's performance. Although a universal phenomenon across all language models was not observed, the findings suggest that language has a discernible impact on basic numeracy. However, the results should not vary with the use of different languages.

5.3 Observation with GPT-4

Table 5 indicates that GPT-4 outperforms all other models, confirming its status as one of the highest-performing LLMs. To ascertain if this observation persists with the optimal model, we examined it with GPT-4, and the results are presented in Table 6. First, it shows a significant difference from the performance of GPT-3.5. Despite variations in rankings, a considerable performance disparity between the best and worst scenarios remains evident. Similarly, the observed reduction in subtraction performance with GPT-3.5 is consistent with our current findings.

6 Conclusion

This study aimed to demonstrate negative results and uncover shortcomings of LLMs in basic arithmetic tasks. Our findings reveal that (1) numeracy is intertwined with linguistic elements, (2) LLMs exhibit suboptimal performance in computation verification tasks, and (3) the concept of carrying/borrowing is not effectively mastered by LLMs, especially borrowing. These results provide a foundation for future research to (1) investigate the robustness of numeracy in language models, (2) enhance computational verification capabilities in number-aware fact-checking tasks, and (3) improve the fundamental arithmetic proficiency of LLMs. Additionally, our observation that language would enhance numeracy is another promising topic that future studies can pay attention to. For example, researchers could investigate how incorporating language-based strategies into mathematics problem-solving improves models' understanding and retention of numerical concepts.

Limitations

This study has two primary limitations. First, due to the vast number of existing LLMs, it is challenging to include all in our analysis. Therefore, we focus on two recent high-performing LLMs: GPT-3.5 and PaLM-2. GPT-3.5 incorporates human feedback during its training, while PaLM-2 relies exclusively on open-source data. We posit that the results obtained from these models on an extensive test set are indicative of general trends. However, future research could employ our proposed test set to compare and analyze additional LLMs. Second, our investigation does not encompass the full spectrum of arithmetic capabilities but is confined to two fundamental operations: addition and subtraction. We encourage subsequent studies to extend our methodology to examine other arithmetic operations. Third, basic arithmetic can actually be solved by generating codes or using additional tools, such as calculators. However, this is beyond the scope of this paper. As shown in Figure 1, some studies utilize LLMs for calculations. Our results show that the performance on the same question may vary when only the language is changed. Moreover, as numbers increase in size, relying on LLMs for arithmetic may not be the best choice. Our findings underscore the importance of using supplementary tools in conjunction with LLMs, and future work could explore more in-depth topics based on our observations.

Acknowledgements

This paper is based on results obtained from a project JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO). The work of Chung-Chi Chen was supported in part by JSPS KAKENHI Grant Number 23K16956.

References

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. The reversal curse: Llms trained on" a is b" fail to learn" b is a". *arXiv preprint arXiv:2309.12288*.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*.

Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2019. Numeracy-600K: Learning numeracy for detecting exaggerated information in market comments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6307–6313, Florence, Italy. Association for Computational Linguistics.

Chung-Chi Chen, Hiroya Takamura, Ichiro Kobayashi, and Yusuke Miyao. 2023a. Improving numeracy by input reframing and quantitative pre-finetuning task. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 69–77, Dubrovnik, Croatia. Association for Computational Linguistics.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. 2023b. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*.

Ernest Davis. 2024. Mathematics, word problems, common sense, and artificial intelligence. *Bulletin of the American Mathematical Society*.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. 2024. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, 36.

Vedant Gaur and Nikunj Saunshi. 2023. Reasoning in large language models through symbolic math word problems. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5889–5903, Toronto, Canada. Association for Computational Linguistics.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14852–14882, Toronto, Canada. Association for Computational Linguistics.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, et al. 2023. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*.

Nabil Hossain, John Krumm, Michael Gamon, and Henry Kautz. 2020. Semeval-2020 task 7: Assessing humor in edited news headlines. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 746–758.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. MathPrompter: Mathematical reasoning using large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 37–42, Toronto, Canada. Association for Computational Linguistics.

Model	Overall	Addition	Subtraction
LLama2-7B	9.38%	11.43%	7.33%
LLama2-7B-Chat	5.17%	5.86%	4.48%

Table 7: Performances of LLama2-7B.

Tiedong Liu and Bryan Kian Hsiang Low. 2023. Goat: Fine-tuned llama outperforms gpt-4 on arithmetic tasks. *arXiv preprint arXiv:2305.14201*.

Piotr Milkowski, Marcin Gruza, Kamil Kanclerz, Przemyslaw Kazienko, Damian Grimling, and Jan Kocon. 2021. Personal bias in prediction of emotions elicited by textual opinions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 248–259, Online. Association for Computational Linguistics.

Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. 2023. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Somin Wadhwa, Silvio Amir, and Byron Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15566–15589, Toronto, Canada. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Noam Wies, Yoav Levine, and Amnon Shashua. 2023. Sub-task decomposition enables learning in sequence to sequence tasks. In *The Eleventh International Conference on Learning Representations*.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing. 2023. Sentiment analysis in the era of large language models: A reality check. *arXiv preprint arXiv:2305.15005*.

A LLama2-7B

Table 7 shows the results of LLama2-7B (Touvron et al., 2023). However, the performance is not as good as the models we discussed, and thus, we did not make discussions based on it.

B Dataset

The dataset is available on the Huggingface[3]. Please note that we control the leading digit to answer other research questions. Thus, the leading digits of two given numbers are always the same. More data can be generated by using the same code.[4]

[3] https://huggingface.co/datasets/NLPFin/BasicArithmetic

[4] https://drive.google.com/file/d/1WahChtYNj4wYy59gYDkvSThFzgQSN7Zh/view?usp=sharing