Studying the capabilities of Large Language Models in solving Combinatorics Problems posed in Hindi

Yash Kumar and Subhajit Roy

Indian Institute of Technology Kanpur, India
{yashk, subhajit}@iitk.ac.in

Abstract

There are serious attempts at improving the mathematical acumen of LLMs in questions posed in English. In India, where a large fraction of students study in regional languages, there is a need to assess and improve these state-of-the-art LLMs in their reasoning abilities in regional languages as well. As Hindi is a language predominantly used in India, this study proposes a new dataset on mathematical combinatorics problems consisting of a parallel corpus of problems in English and Hindi collected from NCERT textbooks. We evaluate the "raw" single-shot capabilities of these LLMs in solving problems posed in Hindi. Then we apply a chain-of-thought approach to evaluate the improvement in the abilities of the LLMs at solving combinatorics problems posed in Hindi. Our study reveals that while smaller LLMs like LLaMA3-8B shows a significant drop in performance when questions are posed in Hindi, versus questions posed in English, larger LLMs like GPT4-turbo shows excellent capabilities at solving problems posed in Hindi, almost at par its abilities in English. We make two primary inferences from our study: (1) large models like GPT4 can be readily deployed in schools where Hindi is the primary language of study, especially in rural India; (2) there is a need to improve the multilingual capabilities of smaller models. As these smaller open-source models can be deployed on not so expensive GPUs, it is easier for schools to provide these models to the students, and hence, the latter is an important direction for future research.

1 Introduction

Large Language Models (LLMs) have revolutionized the technological landscape, with newer applications emerging each day. One of the prime benefactors of this revolution has been the education sector. While initially these models were used as a large knowledge base for facts, the recent models also excel at reasoning tasks like programming and mathematics. This has benefited a large class of students who are using these models as a "personalized tutor" to understand their course material.

These language models are essentially trained over a large corpus of text across the breadth of the internet—online books, wiki articles, blogs, code repositories—to capture the essence of human knowledge. However, most of the text available on the internet is in English. In a country like India, 68.83%(cen) of the population is rural, who predominantly communicate in Hindi and other regional languages. In fact, more than 58%(nue) of the population undergo their school education in the regional languages. Even prestigious exams like IIT-JEE is conducted in thirteen languages, namely English, Assamese, Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Odia, Punjabi, Tamil, Telugu, and Urdu. Hence, it raises an important question the regional language speaking population of the country is equally benefited by the LLMs as the urban, English-speaking population. Or, is the emergence of LLMs increasing the chasm between the Indian population that is being educated in English and other regional languages.

In this work, we investigate the effectiveness of large language models at solving questions presented to them in Hindi, and compare its effectiveness at handling the same problems in English. We use the NCERT textbooks (nce) for the English and Hindi to collect Mathematics questions in the area of Combinatorics. We use multiple strategies and prompting techniques to study the gap in the capabilities of the LLMs at solving mathematical problems posed in these two languages. We conduct this study on three popular models: GPT–3.5 (Radford, 2018), GPT–4(Achiam et al., 2023) and LLaMA3-8B(lla). The reason for selecting these models were that the chat interface of GPT–3.5 is now available freely, making it the most accessible

Proceedings of the First Workshop on Natural Language Processing for Indo-Aryan and Dravidian Languages (IndoNLP2025), pages 90–99
January 20, 2025. ©2025 Association for Computational Linguistics

model for students. GPT–4 is a superior model, but is available against a small monthly fees, and so, is reasonably accessible to students. LLaMA3-8B is a small “open" model that can be run on not-very-expensive GPUs; hence, we believe that soon, schools may decide to host such models within their premises for their students.

We made the following inferences from our study:

- There is a decline in the accuracy of LLMs when it comes to solving problems in Hindi versus English.

- Using different prompting strategies we showed the difference in the performance of the LLMs. "Manual Subcategory" performs better as compared to the other two strategies by upto 14 percent in overall study of Cobinatorics.

- LLaMA3 and GPT-3.5 outperformed themselves when Chain-of-thought prompt strategy is used as compared to the One-shot by a margin of 5 percent for collectively for both the languages.

- LLaMA3-8B and GPT-3.5 showed a significant increase in performance when prompted with an Chain-of-thought in subcategorical analysis by that LLM.

- The above prompt strategy outperformed the other two strategies in 3-4 subcategory cases by a factor of 0.5 to 5 for both the languages.

- GPT-4, being the latest and largest model among others in our studied, outperformed both other models.

This work makes the following contributions:

- We formulate a study to understand the gap in the mathematical abilities of popular open-source models;

- We create a dataset of parallel set of questions in English and Hindi;

- We attempt multiple prompting techniques, single-shot and chain-of-thought prompts and study the improvement in inference accuracy.

- We draw relevant inferences from our study.

In the future, we intend to broaden the scope of this study to more languages, more models and more prompting strategies.

2 Overview

In this work, we attempt to study the following research questions:

- Does posing questions in Hindi as effective as posing the same question in English with single-shot prompts?

- Can inference accuracy be improved with chain-of-thought prompting where the LLM infers the problem subcategory before solving a problem?

To conduct our analysis, we create our own dataset sourcing problems in the area of *Combinatorics* from higher secondary mathematics NCERT textbook (nce) in Hindi and English languages. The dataset contains total of 100 problems in English sourced from English version of the NCERT book and their corresponding parallel counterparts in Hindi sourced from Hindi version of the NCERT book. These problems can be categorised into five subcategories: *Fundamental principle of Counting*, *Permutation with restrictions*, *Permutation without restrictions*, *Combination with restrictions*, and *Combination without restrictions*. The distribution of problems in these subcategories are shown in Table 1.

Table 1: Number of samples in each subcategory

Sub-Category of the Problem	Number of Samples
Fundamental principle of Counting	16
Permutation with restrictions	31
Permutation without restrictions	11
Combinations with restrictions	31
Combinations without restrictions	10

Figure 1 shows an instance from our dataset, consisting of the English and Hindi versions of the problem, its subcategory being "Fundamental principle of counting" and the solution to the problem as "8".

We conducted experiments on three well known large language models: LLaMA3-8B, GPT-3.5 Turbo-175B and GPT-4 Turbo. We used the API calls for the inference of LLaMA3, and chat version of GPT-3.5 Turbo and GPT-4 Turbo for our experimentation . We conduct all experiments on NVIDIA RTX A4000 GPUs. As the responses of the LLMs are sampled from a distribution, we execute each prompt thrice: if any of the answers is

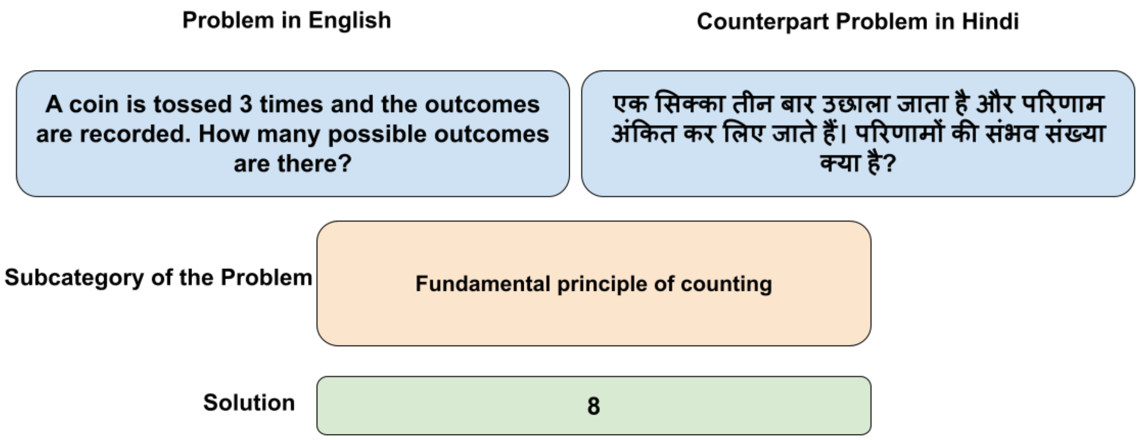

Problem in English
Counterpart Problem in Hindi
A coin is tossed 3 times and the outcomes are recorded. How many possible outcomes are there?
एक सिक्का तीन बार उछाला जाता है और परिणाम अंकित कर लिए जाते हैं। परिणामों की संभव संख्या क्या है?
Subcategory of the Problem
Fundamental principle of counting
Solution
8

Figure 1: Sample problem from our dataset

correct, we mark the problem as solved successfully.

We prompt the LLMs via two prompting strategies: (1) a plain one-shot prompt, requesting the LLM to solve the problem, and (2) a chain-of-thought prompt that asks the LLM to infer the subcategory, and then asks it solve the problem given the subcategory. We discuss this in the subsequent section.

3 One-shot Prompting

In this set of experiments, we prompted the large language models to solve the provided problem. The prompt instructions remain the same for English and Hindi, and only the problem statement is provided in the chosen language. We show an example of the prompts used in Figure 4.

Figure 2 (without the hashed bars) shows the performance of the LLM models for English versus Hindi. There indeed seems to be a chasm between the performance of English versus Hindi, especially for the smaller LLaMA3-8B model. All the LLMs show a decline in accuracy when prompted for Hindi problems as compared to the English problems. The overall difference between the accuracy of English and Hindi problems ranges from 8 percent to 14 percent across all LLMs. The smallest variation in the accuracies if for the case of GPT-4 and highest variation is observed in GPT-3.5.

4 Chain-of-thought Prompting

In this strategy, we apply the following steps:

- We prompted the LLMs to identify the category of the problem out of the given 5 subcategories;
- We prompt the LLM, requesting it to solve the problem *while providing the subcategory*.

A sample prompt given to the LLMs in this stage is given in Figure 1.

4.1 Overall performance

The hashed stacked bars in Figure 2 shows the increase in the accuracy of inference for this prompting strategy versus the single-shot prompting (discussed in Section 3). This prompting strategy does improve the solving capabilities of the LLMs, especially for the smaller LLaMA3-8B model. The overall accuracy increase we found was in the range of 1 percent and 5 percent across all LLMs. LLaMA3-8B shows the highest jump in the accuracy: 5 percent for English problems and 3 percent for Hindi problems using the Chain-of-thought prompt. Another high variation in accuracy can be seen in GPT-3.5 case for Hindi problems where we got an increase of 4 percent.

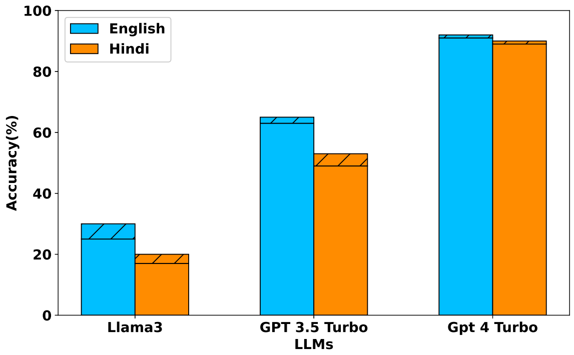
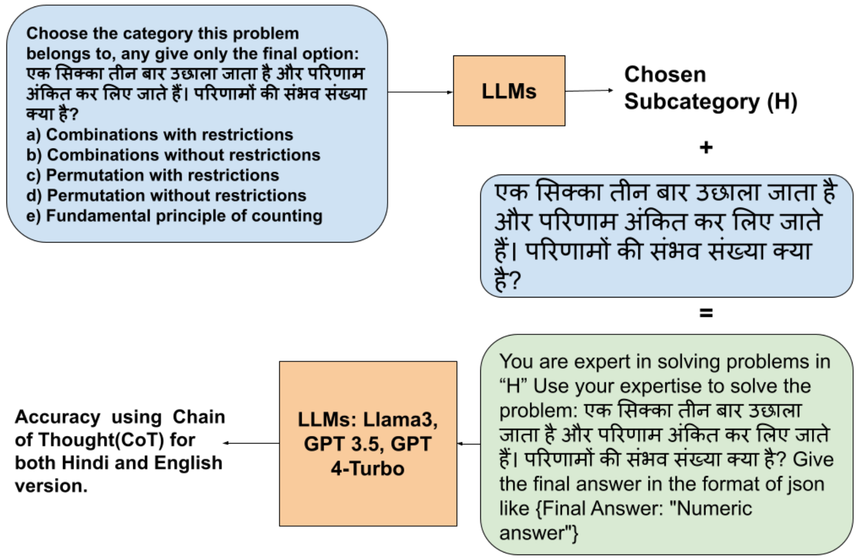


Figure 2: Comparison of One Shot and Chain-of-thought prompt strategies applied on English and Hindi problems

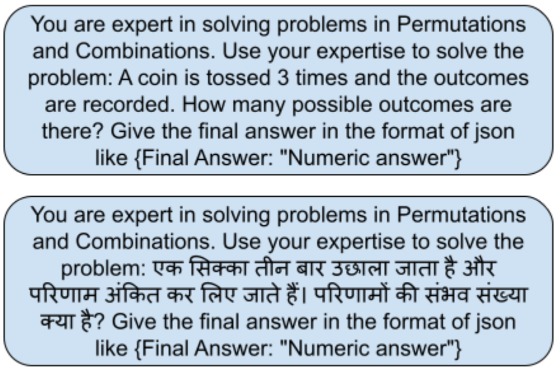

LLMs
Chosen Subcategory (H)
+
=
LLMs: Llama3, GPT 3.5, GPT 4-Turbo
Accuracy using Chain of Thought(CoT) for both Hindi and English version.

Choose the category this problem belongs to, any give only the final option:
एक सिक्का तीन बार उछाला जाता है और परिणाम अंकित कर लिए जाते हैं। परिणामों की संभव संख्या क्या है?
a) Combinations with restrictions
b) Combinations without restrictions
c) Permutation with restrictions
d) Permutation without restrictions
e) Fundamental principle of counting

एक सिक्का तीन बार उछाला जाता है और परिणाम अंकित कर लिए जाते हैं। परिणामों की संभव संख्या क्या है?

You are expert in solving problems in "H" Use your expertise to solve the problem: एक सिक्का तीन बार उछाला जाता है और परिणाम अंकित कर लिए जाते हैं। परिणामों की संभव संख्या क्या है? Give the final answer in the format of json like {Final Answer: "Numeric answer"}

Figure 3: Inference using the Chosen Subcategory by LLM



You are expert in solving problems in Permutations and Combinations. Use your expertise to solve the problem: A coin is tossed 3 times and the outcomes are recorded. How many possible outcomes are there? Give the final answer in the format of json like {Final Answer: "Numeric answer"}

You are expert in solving problems in Permutations and Combinations. Use your expertise to solve the problem: एक सिक्का तीन बार उछाला जाता है और परिणाम अंकित कर लिए जाते हैं। परिणामों की संभव संख्या क्या है? Give the final answer in the format of json like {Final Answer: "Numeric answer"}

Figure 4: Prompt used in One Shot Prompt

Then, in the second stage, the LLMs were prompted to choose the subcategory that the problem belonged to given in Figure 3. Here also, we had total 100 prompts for each language. Only one trial was run acrosss all LLMs. After the successful completion of this setting we obtained the Chosen Subcategories: **E** for English version and **H** for Hindi version of the problem. We used only the this H for the overall analysis and accurately chosen subcategories for subcategorical analysis below further in our pipeline. Lastly, we experimented with the actual subcategories-Manual Subcategory. The prompts which we designed here were again used in the inference of all the three LLMs for three trails each. The bar plots mentioned in Section 4.2, show an increase in performance for 3-4 categories when using Chain-of-thought prompting strategy and Manual Subcategory also shows an increase in accuracy as compared to the One-Shot prompt strategy. The subcategorical analysis is discussed in detail in Section 4.2.

4.2 Detailed analysis by subcategories

Table 2 shows the accuracy of LLMs in choosing or assigning the correct subcategory out of the 5 choices given to them. Here, as expected GPT-4 performs better as compared to GPT-3.5 and LLaMA3 in classifying the given problem, be it in Hindi or English, with its associated subcategory. In most of the subcategories, we observed GPT-4 Turbo performing well in assigning the subcategories with an exception in Fundamental principle of Counting category in English problems and Combination without restriction in Hindi problems. LLaMA3 performed lowest among all the three LLMs in this task with an exception in case of Fundamental principle of Counting subcategory where it outperformed both GPT-versions.

Now, we discuss about the LLMs performance in each subcategory across English and Hindi problems using three prompting strategies: "One-shot", "Chain-of-thought" and "Manual Subcategory". Please refer to Table 4 for finding the full name of subcategory mentioned in the bar plots. From Figure 7, we can infer that the cases where we used Chain-of-thought prompt strategy, the performance increases by a factor starting from $\mathbf{0.42}$ to as high as $\mathbf{4.92}$ times when compared with One-

Table 2: LLMs' Accuracy for choosing Question's Sub-Category

Sub-Category	**LLMs**	**Question in English**	**Question in Hindi**
Fundamental principle of Counting	LLaMA3	81.25	81.25
	GPT-3.5	12.5	31.25
	GPT-4	18.75	43.75
Permutation with restrictions	LLaMA3	22.58	0
	GPT-3.5	61.29	9.27
	GPT-4	87.09	80.64
Permutation without restrictions	LLaMA3	18.18	0
	GPT-3.5	18.18	0
	GPT-4	45.45	54.54
Combination with restrictions	LLaMA3	3	6.45
	GPT-3.5	61.29	19.25
	GPT-4	83.87	74.19
Combination without restrictions	LLaMA3	10	0
	GPT-3.5	20	30
	GPT-4	20	20

shot prompt strategy when both language cases are taken collectively. There are exception cases of 2 subcategories in English version where the performance is almost the same as observed in the One-shot prompt strategy. For Hindi case, LLaMA3 couldn't solve any sample for Subcat 4 and Subcat 5. Also, in 3-4 subcategories in both the languages, we see an increase in the performance of Chain-of-thought prompt strategy when we compare with the subcategory prompt strategy by a factor of 1.2 to 3.2 times. It is worth mentioning the results we observed when using subcategory prompt strategy, where we got an increase in performance from One-shot prompt strategy by a factor of 1.42 to 4 times. There are cases where it showed similar performance as that of One-shot prompt strategy and an exception of 1 category with low performance than One-shot.

Similarly, from Figure 8, in English language we see an increase in performance while using Chain-of-thought prompt strategy over the other two strategies in 4 subcategories. For Hindi case, we see either similar or more performance in 3 subcategories for Chain-of-thought prompt strategy. The performance increase that we observed ranged from 1.11 to 2 times for English case and 1.06 to 2 times for Hindi case. If we compare the cases where we used subcategories for prompting, we got a performance increase of 1.14 to 1.25 times for English case and 1.33 to 1.73 times for Hindi problems. If we look at Figure 9, we observed almost similar performance in all three strategies. There was an exception of Subcat 1, 2 and 3 where Chain-of-thought outperformed the One-shot prompt in both the languages. The subcategory prompt strategy was also similar to the other two. Given the fact that GPT-4 is the latest and largest model in our study, the result obtained is expected.

Table 3: Accuracy of LLMs in identiying the Subcategories

Model	English	Hindi
LLaMA3	24	15
GPT-3.5	46	17
GPT-4	63	63

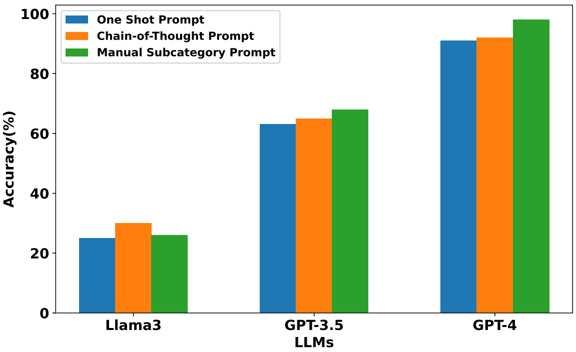


Figure 5: Results using different prompt strategies on English problems

5 Error Analysis

The performance of this scheme depends on the following factors:

Table 4: Name of abbreviations used in bar plots

Sub-Category	Name of the abbreviated Subcategory
Subcat 1	Fundamental principle of Counting
Subcat 2	Permutation with restrictions
Subcat 3	Permutation without restrictions
Subcat 4	Combinations with restrictions
Subcat 5	Combinations without restrictions

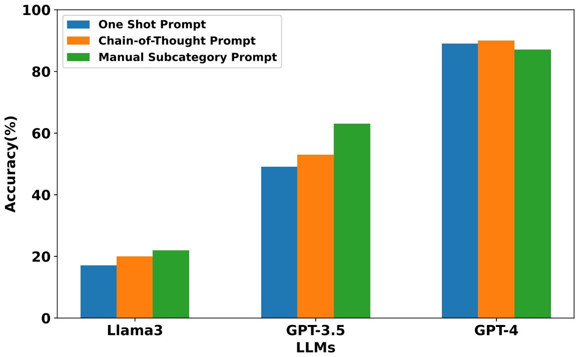


Figure 6: Results using different prompt strategies on Hindi problems

1. The *understanding of Hindi* language by the Large Language Models, i.e. how well LLaMA, GPT-3.5 and GPT-4 understand Hindi? (Task 1)

2. The accuracy of the classification into subcategories, i.e. does the LLM classify the problems into the right subcategories? (Task 2)

3. The accuracy of problem solving *once the subcategory is provided.* (Task 3)

For task 1, we utilized Hindi comprehension problems derived from NCERT textbooks (nce) to evaluate the performance of large language models (LLMs). Specifically, we curated a dataset comprising ten passages in Hindi, each accompanied by five corresponding questions. These passages and questions were directly provided as prompts to the LLMs to assess their accuracy on this task. Our results indicate that LLaMA3-8B achieved an accuracy of 50%, whereas GPT-3.5 and GPT-4 Turbo both attained 76% accuracy. These findings highlight the superior proficiency of GPT-3.5 and GPT-4 Turbo in understanding Hindi compared to the smaller LLaMA3 model. This also concludes the similar trends observed in task involving combinatorics problems framed in Hindi, further corroborating the relative strengths of GPT-based models in processing the Hindi language.

Table 3 studies the accuracy for the subcategory classification task 2. As can be seen, the accuracy of identifying the problem type is low. However, the language models are more accurate in choosing the subcategory of the problem given in English compared to the same problem in Hindi which we can conclude from the results obtained from task 1.

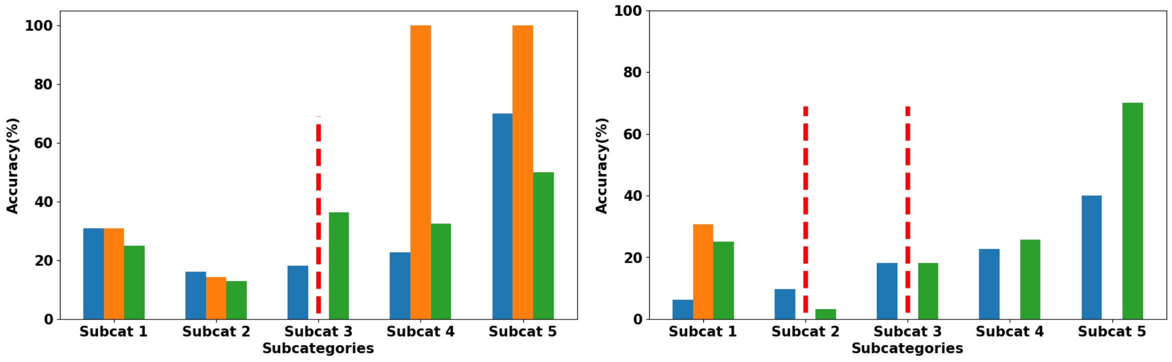
To further understand the impact of this on Combinatorics problems, we ran another set of experiments in task 3 where we manually provided these subcategories within the prompt. The first two bars in the plots 5 and 6 show the solving accuracy corresponding to one-shot and chain-of-thought prompting (for English and Hindi, respectively). The third bar shows the accuracy of the end-to-end pipeline for solving the mathematical problems if the subcategory is provided (*manually*) within the prompt; we refer to this as “Manual Subcategory". We highlight the inference of the performance of language models on problems posed in English with chain-of-thought prompts and manual subcategory prompts from the second and third bar of the plot 5 after the results obtained in task 2.

Interestingly, LLaMA3-8B provides a curious case: though its subcategory inference accuracy is low, the inference accuracy of the end-to-end pipeline increases with chain-of-thought prompting. Still more strangely, its accuracy drops if we manually provide the right categories for English problems. We are still trying to understand this counter-intuitive behavior from LLaMA3-8B.

6 Related Work

In this section, we will discuss about any recent works related to our LLMs solving mathematical problems in English. To the best of our knowledge, there is currently a lack of research on improving the mathematical capabilities of LLMs in regional languages.

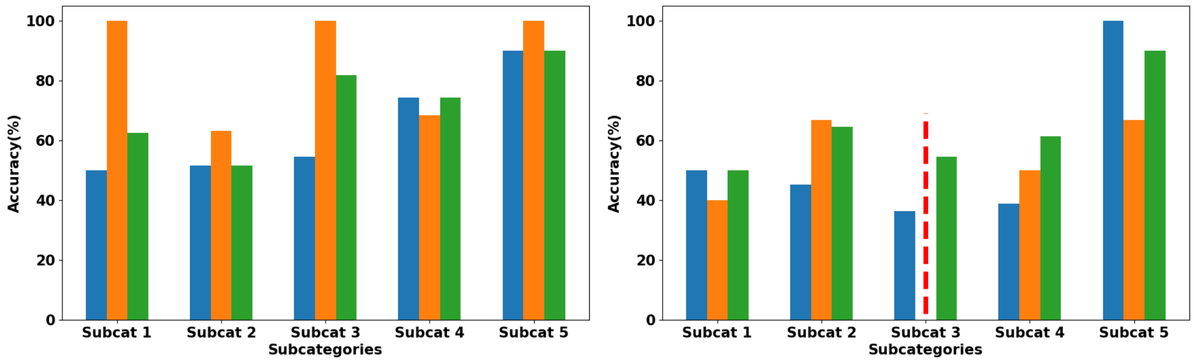
Attempts have been made to improve the math-



(a)

(b)

Figure 7: (a) LLaMA3-8B performance in three strategies for English problems, (b) LLaMA3 performance in three strategies for Hindi problems: **Orange bars: Chain-of-thought prompt strategy**, **Green bars: Manual Subcategory Prompt Strategy** and **Blue bars: One-Shot Prompt Strategy**. **Red line** shows there are no samples/problems for which LLM chose subcategory accurately.

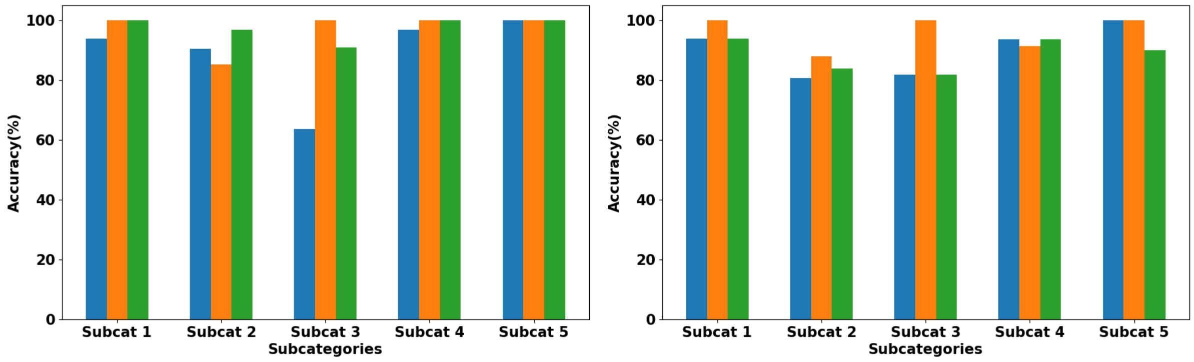


(a)

(b)

Figure 8: (a) GPT-3.5 Turbo performance in three strategies for English problems (b) GPT-3.5 Turbo performance in three strategies for Hindi problems: **Orange bars: Chain-of-thought prompt strategy**, **Green bars: Manual Subcategory Prompt Strategy** and **Blue bars: One-Shot Prompt Strategy**. **Red line** shows there are no samples/problems for which LLM chose subcategory accurately.

ematical capabilities of LLMs in solving mathematical word problems in English language. Recent works highlight the performance of LLMs in English mathematical word problems. A major part of advances in the area started with the design of datasets for math word problems in English, (Frieder et al., 2024) is one such work where miniGHOSTS and GHOSTS are extracted from publicly available datasets and were used to analyse the abilities of ChatGPT-3.5 and 4. (Srivastava and Kim, 2024) proposes a strategised version of masking during pre-training stage of Encoder-Decoder models instead of random masking which significantly improved the performance of Encoder-Decoder small scale models by 2-3 times on benchmark mathematical datasets (English). A special method, MathPrompter(Imani et al., 2023), enhances arithmetic operations and reasoning capabilities of LLMs leveraging the programming capabilities of LLMs as an intermediate step in solving the problem. They worked on english word problems dataset (Roy and Roth, 2015) and showed an improved performance by almost 15%. Mathify(Anand et al., 2024), another recent study in this area, where they sourced a mathematical word problem dataset, named MathQuest, from the English NCERT textbook. Using this dataset they fine-tuned open source large language models and compared their performance. Another work (Wei et al., 2022), uses the Chain of Thoughts prompt strategy on LaMDA(Thoppilan et al., 2022) and PaLM(Chowdhery et al., 2023) and showed almost 100ing accuracy on GSM8K(Cobbe et al., 2021). (Chen et al., 2023) used Program of Thoughts strat-



(a)

(b)

Figure 9: (a) GPT-4 Turbo performance in three strategies for English problems (b) GPT-4 Turbo performance in three strategies for Hindi problems: **Orange bars: Chain-of-thought prompt strategy**, **Green bars: Manual Subcategory Prompt Strategy** and **Blue bars: One-Shot Prompt Strategy**

egy instead of Chain of Thoughts, just like Math-Prompter discussed above to improve LLMs' performance on numerical tasks. It compared the CoT methods and PoT meth- ods, resulting in the PoT method outperforming the CoT method in solving numerical problems.

Some works targeting multilingual tasks include xSTREET(Li et al., 2024), which targets to improve the reasoning capabilities including but not limited to mathematics of LLMs across non-English languages: Arabic, Spanish, Russian, Chinese, and Japanese. Here, they leveraged the reasoning capabilities of LLMs trained on code or programs, which they claim that are good reasoners from their study as compared to the LLMs trained on non-code data. ConceptMath(Wu et al., 2024), another study that targets to analyse and comprehend the LLMs in mathematical reasoning tasks in English and Chinese. They did this study on elementary and middle school level mathematical data. Their study focuses ore on the granules of mathematics like statistic, geometry, etc. instead of studying mathematical work problems as a whole. This study contributed to improvement part using an efficient fine-tuning setting where post their analysis on granular level, they used benchmark datasets like MATH(Hendrycks et al., 2021) and GSM-8K(Cobbe et al., 2021) along with their data, to fine-tune the LLM to improve it's performance in that mathematical area. (Le et al., 2024) uses chain-of-thought technique with high-quality in-context learning exemplars obtained by multilingual dense retrieval to enhance LLM's performance in mathematics.

7 Supplementary Materials

We encourage readers to review the prompts used and datasets created for this study. The access to the datasets developed and the prompts used to carry out this study is given in this github link:[1]. The supplementary materials accompanying this paper include a folder named Datasets which includes three CSV files, one for each of the language models evaluated in the study, containing problems in permutations and combinations presented in both English and Hindi. There is prompts file having the prompts used to generate the responses from LLMs. Furthermore, these prompts can be utilized to interface with the language models. These resources are provided to ensure transparency, reproducibility, and ease of future research based on our findings.

8 Conclusions and Future Work

Our main focus of study was analysing the performance of LLMs in solving combinatorics problems in Hindi so as to assess them, if they can be readily deployed in the education sector. For our study, we used GPT-3.5, a freely available LLM with a chat interface; LLaMA3-8B, a small "open" source model that can be run on an affordable GPU, and GPT-4 Turbo, one of the most powerful models available currently. In future research, we plan to significantly expand our dataset to encompass over 100 problems per subcategory, aiming to improve both its comprehensiveness and robustness. This effort will facilitate a deeper exploration of mathematical problem-solving across diverse categories,

[1] https://github.com/yash-raj-verma/IndoNLP_COLING_2025.git

ensuring more representative benchmarks. Furthermore, we will broaden the linguistic scope of our study by incorporating additional Indian regional languages, such as Bengali, Tamil, Assamese, and Urdu, alongside non-Indian languages, including Greek and Arabic. This expansion will enable a cross-cultural examination of mathematical reasoning and problem formulation in various linguistic contexts.

To further enhance the scope and impact of our work, we intend to evaluate the capabilities of emerging state-of-the-art language models on our enriched datasets. By incorporating models with improved architectures and training paradigms, we aim to uncover new insights into their generalization and adaptability. Additionally, we plan to use our dataset for fine-tuning smaller, efficient models, such as LLaMA3, with a focus on exploring their potential for targeted improvements in performance, particularly in resource-constrained environments. This dual approach promises to deepen our understanding of model behavior while driving innovation in both large-scale and lightweight language model applications. We believe that such studies would benefit a country like India or others (once the analysis and scope of this work expands to other regions and their regional languages), where there exists a large number of regional languages in which education is imparted, and show the way forward for LLMs effective currently for all segments of the Indian population with the intention of expanding this to other countries.

Limitations

While our research investigates the application of large language models (LLMs) to solving mathematical problems in Hindi, certain limitations persist. One significant constraint is the size and scope of our dataset, which comprises only 100 problems per subcategory. This limited sample may hinder the robustness and comprehensiveness of our evaluation. Expanding the dataset to encompass a wider range of problems, drawn from additional mathematical topics or diverse educational resources such as textbooks in other languages, would help enhance its representativeness and reliability.

Moreover, our study is centered on evaluating the performance of LLMs, but it does not explore the potential benefits of fine-tuning smaller, more resource-efficient models on the same dataset. Investigating the performance improvements achievable with such fine-tuning could provide valuable insights into balancing computational efficiency with model accuracy.

To address these limitations, future work would prioritize not only the expansion of the dataset to include a richer variety of problem types but also the exploration of smaller, fine-tuned models. This dual approach could increase the diversity of the mathematical problems handled while also improving the accessibility and scalability of our study, particularly for educational settings with limited computational resources and diverse linguistic backgrounds.

References

Census 2011. https://web.archive.org/web/20180127163347/http://planningcommission.gov.in/data/datatable/data_2312/DatabookDec2014%20307.pdf.

Meta llama team. introducing meta llama 3: The most capable openly available llm to date. (accessed on this url). https://ai.meta.com/blog/meta-llama-3/.

National council of educational research and training. https://ncert.nic.in/textbook.php.

National institute of educational planning and administration. https://niepa.org.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Avinash Anand, Mohit Gupta, Kritarth Prasad, Navya Singla, Sanjana Sanjeev, Jatin Kumar, Adarsh Raj Shivam, and Rajiv Ratn Shah. 2024. Mathify: Evaluating large language models on mathematical problem solving tasks. *arXiv preprint arXiv:2404.13099*.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

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. *CoRR*, abs/2110.14168.

Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Petersen, and Julius Berner. 2024. Mathematical capabilities of chatgpt. *Advances in neural information processing systems*, 36.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models. In *ACL (industry)*, pages 37–42. Association for Computational Linguistics.

Nguyen-Khang Le, Dieu-Hien Nguyen, Dinh-Truong Do, Chau Nguyen, and Minh Le Nguyen. 2024. Vietnamese elementary math reasoning using large language model with refined translation and dense-retrieved chain-of-thought. In *JSAI International Symposium on Artificial Intelligence*, pages 260–268. Springer.

Bryan Li, Tamer Alkhouli, Daniele Bonadiman, Nikolaos Pappas, and Saab Mansour. 2024. Eliciting better multilingual structured reasoning from LLMs through code. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5154–5169, Bangkok, Thailand. Association for Computational Linguistics.

Alec Radford. 2018. Openai gpt paper titled improving language understanding by generative pre-training.

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.

Nilesh Srivastava and Seongchan Kim. 2024. Enhancing mathematical reasoning in math word problems: A numerical masking approach for encoder-decoder models. *Elsevier BV*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

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.

Yanan Wu, Jie Liu, Xingyuan Bu, Jiaheng Liu, Zhanhui Zhou, Yuanxing Zhang, Chenchen Zhang, Zhiqi Bai, Haibin Chen, Tiezheng Ge, et al. 2024. Conceptmath: A bilingual concept-wise benchmark for measuring mathematical reasoning of large language models. *arXiv preprint arXiv:2402.14660*.