EVALUATING THE ROBUSTNESS OF RSTAR: A NOVEL FRAMEWORK FOR ENHANCED REASONING IN SMALL LANGUAGE MODELS

Jone Steinhoff [s243867], *Lukas Rasocha* [s233498], *Panagiota Emmanouilidi* [s223531], *Petr B. Nylander* [s240466], *Robert Spralja*[s243658]

Supervised by Prof. Ole Winther and Davide Venuto
DTU Compute, Technical University of Denmark

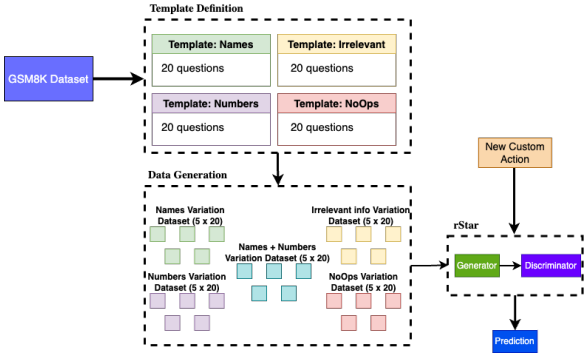

Fig. 1: Our project overview: First the GSM8K dataset was subsampled. Then multiple of generic templates were designed, each capable of automatically generating different altering variations of the original GSM8K subsampled questions. The resulting synthetically created dataset was used to test the robustness of the *rStar* framework, where we also experimented with new custom actions added to the existing set of actions proposed by the original work [1]. (Note: NoOps variations were designed manually).

ABSTRACT

This study evaluates the robustness of the *rStar* framework, a reasoning system for small language models (SLM), using experiments that include varied input variations of a subset from the GSM8K dataset. The experiments consist of changes to names, numerical values, irrelevant information, and additional noncontributory data (NoOps). Findings revealed that *rStar* showed robustness to changes such as name replacements but struggled with numerical changes, irrelevant information, and NoOps where accuracy dropped significantly. We have also experimented with adding a new custom action, designed to filter out irrelevant information in the question and thus counter the *irrelevant information* variations. Results showed that in most cases this action led to the removal of important details and therefore leading to incorrect answers. All the code used for this project is available in our github. [1]

[1]https://github.com/lukyrasocha/rStar

1. INTRODUCTION

The **S**elf-play mu**T**u**A**l **R**easoning (*rStar*) framework claims to show promising results in handling reasoning tasks, but its adaptability to variations in problem structures need further exploration [1]. We investigate the robustness of *rStar* by introducing modifications inspired by the GSM-Symbolic approach [2] to test its performance on the Grade School Math 8K (GSM8K) dataset. While SLMs exhibit strong results on GSM8K, they may overestimate their true reasoning abilities due to potential data contamination. Specifically, parts of the test set may overlap with the training data, giving models prior knowledge of certain questions and inflating performance metrics. Prior studies, such as GSM-Symbolic, have revealed that even minor alterations to input, such as numerical changes or the addition of irrelevant elements, can degrade the reasoning performance of models [2]. This motivates a deeper examination of *rStar*'s ability to handle such variations effectively, providing insights into its strengths and limitations. Our methods consists of first producing baseline results on a subset of GSM8K and then introducing 5 experiments with variations including changes to names, numbers, relevant and irrelevant information. The impact of these modifications is analyzed, where also an additional action is incorporated into *rStar*'s existing set, to assess whether such input variations introduced by the experiments can simply be countered by the reasoning actions of the framework. Through this exploration, we aim to evaluate *rStar*'s true reasoning capabilities and its potential to handle more realistic scenarios where inputs are often unpredictable.

2. RELATED WORK

This project was intended to emulate GSM-Symbolic and GSM-NoOp, a dataset developed by Mirzadeh et al. [2], to test the robustness of language models across variations. In their work, they created templates from a subset of 100 GSM8K questions. For example, names, numbers, and other words like relations or sports. After that they randomly assigned values and 50 variations were created for each data

point. The paper concludes that SLMs and LLMs alike show a form of token bias, in that the models perform worse (sometimes several standard deviations) on the variations than the original GSM8K questions. They have also shown that just changing the names and nouns results in a performance drop, and when changing numbers or both names and numbers the performance drop is even higher. Their evaluation used an 8-shot chain-of-thought prompt with greedy decoding.

Another example of such evaluation is the dataset GSM1K from Zhang et al. [3]. In their work, they generated a set of one thousand new problems designed to resemble GSM8K. While these problems also resulted in a decrease in accuracy, the drop was relatively smaller compared to the more significant declines observed in GSM-Symbolic.

Furthermore, Jiang et al. [4] performed another similar analysis, this time on logical problems, and still identified signs of data contamination in terms of token bias.

3. METHODOLOGY

3.1. rSTAR

The *rStar* (a **S**elf-play mu**T**u**A**l **R**easoning) framework consists of two main stages, which collaboratively enhance the reasoning capabilities of SLMs through mutual feedback mechanisms. A diagram of the stages can be seen in Figure 4.

3.1.1. Reasoning Generation Stage

This stage uses a Monte Carlo Tree Search (MCTS) algorithm, introduced by Rémi Coulom in 2007 [5], to guide the first SLM (Mistral 7B) in generating reasoning trajectories. Each node in the tree represents an intermediate reasoning step, while edges correspond to human-like reasoning actions (represented by predefined prompts to the model). These actions include *proposing single-step thoughts, completing remaining steps, breaking questions into sub-questions, revisiting sub-question answers*, or *rephrasing the question for clarity*. [1]

The generator stage constructs a range of reasoning trajectories by exploring this rich action space. The trajectories are evaluated based on a reward function Q . At the end of each trajectory, the SLM generates a set of 10 candidate answers. The confidence is then calculated as the frequency of the most frequent candidate answer divided by 10. The confidence score is then used to update the reward Q , which is subsequently propagated back along the trajectory to inform earlier steps. The generator then uses this information to ensure a balance between exploring new reasoning paths and exploiting high reward actions. [1]

3.1.2. Reasoning Validation Stage

Once candidate reasoning trajectories are generated, the discriminator stage uses another SLM (Phi-3-mini-instruct) to verify their correctness. The discriminator SLM receives partial reasoning trajectories and is asked to complete the missing steps. If the discriminator's completion aligns with the original trajectory, the reasoning path is checked as valid. In the scenario where multiple paths are deemed valid, the path with the highest reward score Q is chosen as the final answer.

This mutual reasoning consistency reflects the peer-review like process, where agreement between two independent parties (models) suggests a higher likelihood of correctness.

3.2. Experimental setup

To test the robustness of the *rStar* algorithm, we first randomly select a small subset of 20 questions from the GSM8K test dataset. This subset is chosen due to the computational constraints, described in Section 5.2. We evaluate *rStar*'s accuracy on this subset to establish a baseline for comparison with the experiments described in Section 3.2.1. For this, we use the code and the exact experiment setup provided by the authors of the original paper [1].

3.2.1. Variations of the GSM8K dataset

Adopting the approach from [2], we enhance the GSM8K dataset by introducing five variation templates for the baseline. These templates are used to generate novel variations of the questions in the baseline dataset. We introduce following templates:

1. **Names**
   We annotate the names in each question with a variable placeholder. Each placeholder gets assigned a randomly selected name from a list of female and male names, depending on the context of the question.

2. **Numbers**
   In this variation, all numerical values are replaced by variable placeholders. Each placeholder is assigned randomly generated integer within a specified range. We include conditions in order to ensure the final question still makes sense.

3. **Names & Numbers**
   This variation is the combination of **Names** and **Numbers**.

4. **NoOps**
   Another variation proposed by [2] includes inserting sentences in theme of the questions. These questions do not contain any numerical values, but contain a contextual information related to the theme of the question. They do not affect, how the questions should be

solved. The goal is to introduce novel information, that the model has not seen during training and thus test its reasoning skills, rather purely rely on pattern matching [2].

5. **Irrelevant Information**
   The last variation is proposed by us. We add a thematically irrelevant sentence containing numerical value to the baseline questions. In contrast to **NoOps**, where the added sentence that was in the theme of the question, but had no impact on the solution. The goal of this variation is to asses the ability of the language model to determine, what information is important for the problem and which sentence can be ignored.

For each of the variations, we create five realizations, resulting in a total of 500 questions. These questions are evaluated with *rStar* and their accuracy is compared against the baseline. All variations were generated in Python, except the NoOps, which was done manually in order to achieve the thematic relevance with the help of ChatGPT [6]. An example of the templates can be seen in Figure 8.

3.3. Action 6: Filter out irrelevant information

After we had finished the experiments, mentioned in Section 3.2, we have encountered that the generator SLM incorrectly used the irrelevant information when producing the reasoning trajectories (see section section 4). Therefore, we propose a new action, which takes the question and tasks the SLM to filter out all irrelevant information to produce a cleaner version of it. The reasoning steps are then performed on this filtered out question. We designed a following prompt one-shot, included in the Appendix A.2. This action was designed specifically to alleviate the complexity introduced by the Irrelevant Information variation described in section 3.2.

4. RESULTS

To investigate *rStar*'s robustness, we evaluated for each experiment the performance of both the discriminator and the majority-vote generator (@maj) by calculating their accuracy scores. These accuracy values were then averaged across all variations introduced within each experiment so that we could compare results of all experiments with the baseline.

Figure 2 summarizes the results of our evaluation, showing the average accuracy of both the discriminator and the majority-vote generator across all experiments. The bars represent the averaged accuracy for each experiment, while the horizontal dashed lines indicate the baseline accuracy levels for the discriminator and generator. The error bars, represent one standard deviation within the results and we have chosen to include them to illustrate the variability in performance across different variations for each experiment.

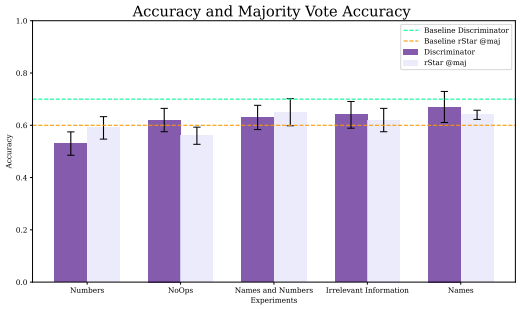

Fig. 2: Average accuracy of the discriminator and rStar @maj across all experiments. The bars indicate the averaged accuracy for each experiment, with dashed lines marking the baseline accuracy levels. The error bars represent one standard deviation within the averaged accuracy.

Figure 2 depicts that the *rStar* framework has varying levels of robustness across the different types of variations introduced in the experiments. The results showed that *rStar* is sensitive to changes in numerical values, as seen in the "Numbers" experiment where accuracy for both the discriminator and majority-vote generator was the lowest among all experiments. This indicates that numerical changes introduce logical complexities that the framework struggles to manage effectively. Additionally, the "Names and Numbers" experiment, the "NoOps", and at the "Irrelevant Information" had a similar accuracy drop of the discriminator compared to the baseline. Only the "Names" experiment showed robustness to changes which shows that rStar is more resilient to variations that do not disrupt the underlying logical structure of the problems. This can also be seen in Figure 3 where the accuracy drop for "Numbers" relative to the baseline is -17%.

In Figure 3 we highlight the accuracy drops ($\%$) experienced by *rStar* relative to the baseline for each of the experiments and furthermore we have also chosen to include two results from Mirzadeh et al. [2].

It can be seen that the "Mistral-7b-v0.1@GSM-NoOp" [2] (Mistral is the generator SLM we use) experiment showed a drop of -28.3%, but our "NoOps" experiment exhibited only a -8% drop. This difference could be related to differences in how the "NoOps" variations were implemented in the two setups. For example, in "Mistral-7b-v0.1@GSM-NoOp" , they likely involved more complex additions to the reasoning path, which could have confused the model more. In contrast, our NoOps implementations might be simpler, resulting in a smaller accuracy reduction.

Moreover, moderate drops in experiments like "Names and Numbers" -7% and "Irrelevant Information" -6% suggest that the model faces some difficulty when dealing with more complex variations. Interestingly, what we didn't expect was for the accuracy drop to be higher for "Numbers" -17% but lower when combined with "Names + Numbers" -7%. This could potentially be due to data contamination, where

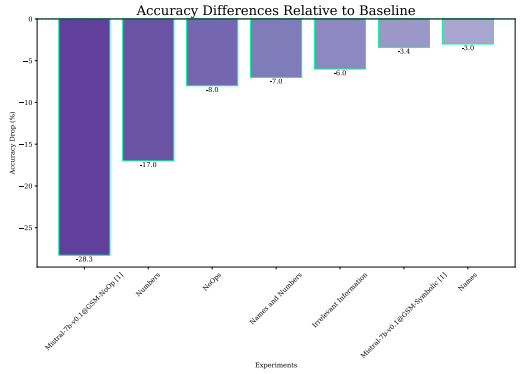

Fig. 3: Accuracy drops (%) relative to the baseline across all experiments, highlighting rStar's sensitivity to different types of variations. The figure also includes results from the "Mistral-7b-v0.1@GSM-NoOp" and "Mistral-7b-v0.1@GSM-Symbolic" experiments from Mirzadeh et al. [2]

we hypothesize that exposure to more randomness forced the model to reason about the problem rather than do simple pattern matching to the potential information seen during training.

Lastly, we tested our new custom Action 6 on one variation of the Irrelevant Dataset. The generator filtered out the irrelevant information only in 2 cases out of 20. In the other cases it failed to filter it out or it removed other information which was necessary for the question to be answered correctly. In one interesting example, the filtering was correct, however the model also erased a part of the relevant information. Despite this, the model still ended up answering correctly, which could further hint data contamination. See examples figure 9 - 11 in the appendix A.6. Nevertheless, to evaluate this action more thoroughly, more testing would be needed.

In summary, the results in Figure 2 and Figure 3 show the varying levels of robustness exhibited by the *rStar* framework across the different experimental variations. While *rStar* struggles with numerical changes, and other types of variations including the "NoOps", the "Irrelevant" and the "Numbers and Names" experiments, it shows to be robust when it comes to variations in names such as in the "Names" experiment. In order to reproduce the Figure 2 and Figure 3, you need to run and inspect the *main.ipynb* notebook included in our github repository.

5. CONCLUSION

This project dealt with testing the robustness of the *rStar* framework by including variations of the GSM8K dataset. We found that all of the experiments had a drop in the *rStar* performance, with all but one dropping beyond one standard deviation (Figure 2). The largest drop could be seen in experiment 2, where the numbers were changed, and the smallest drop was seen in experiment 1, where the names were changed. This indicates that the *rStar* algorithm seems to be less robust to numerical changes in the questions. However, when combined, experiment 1 and experiment 2 have a considerably less drastic drop than experiment 2 alone. A reason for that could be that the data before was contaminated, meaning some part or all of the GSM8K dataset had been used in training our model. This way, it could pattern-match the correct solution. This approach then failed in experiment 2. But when combined with experiment 1, we hypothesize that the model was "shocked" and, instead of pattern-matching, attempted to reason. Apart from the numerical and name changes, experiment 4 (NoOps) and experiment 5 (Irrelevant) also reduced the accuracy compared to the baseline, with experiment 4 resulting in a slightly bigger drop. This suggests that *rStar* handles ignoring irrelevant information better than handling seemingly relevant information.

Overall, *rStar* does not seem robust to the various input variations; however, compared to not using *rStar*, it does reduce the drop in accuracy significantly, as seen in Figure 3, although there might be a difference in complexity of the NoOps datasets.

5.1. Future work

While this project provided a good introduction to evaluating the robustness of a framework like *rstar*, it is clear that this can be expanded upon. One such expansion is the size of the dataset. More variations and a bigger subset of questions could lead to more accurate results. Additionally, the experiments could be executed more times to account for the inherent variance of language models. Furthermore, evaluating Mistral on the NoOps dataset, without using the *rStar* framework, would make a better comparison between our results and the results from Mirzadeh et al. We would also like to further explore Action 6 and experiments with different types of prompts and run it on more variations [2].

5.2. Limitations

We also encountered some limitations that have to be taken into account. The generator alone took approximately 12 hours to run one variation (20 questions) using a Tesla A100 PCIE 40 GB, while the discriminator, with a similar setup, took about 6 hours. These limitations restricted the extent of the experiments.

6. REFERENCES

[1] Zhenting Qi, Mingyuan Ma, Jiahang Xu, Li Lyna Zhang, Fan Yang, and Mao Yang, "Mutual reasoning makes smaller llms stronger problem-solvers," 2024.

[2] Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar, "Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models," 2024.

[3] Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Charlotte Zhuang, Dylan Slack, Qin Lyu, Sean Hendryx, Russell Kaplan, Michele Lunati, and Summer Yue, "A careful examination of large language model performance on grade school arithmetic," 2024.

[4] Bowen Jiang, Yangxinyu Xie, Zhuoqun Hao, Xiaomeng Wang, Tanwi Mallick, Weijie J. Su, Camillo J. Taylor, and Dan Roth, "A peek into token bias: Large language models are not yet genuine reasoners," 2024.

[5] Rémi Coulom, "Efficient selectivity and backup operators in monte-carlo tree search," in *Computers and Games*, H. Jaap van den Herik, Paolo Ciancarini, and H. H. L. M. (Jeroen) Donkers, Eds., Berlin, Heidelberg, 2007, pp. 72–83, Springer Berlin Heidelberg.

[6] OpenAI, "Chatgpt conversation on variations and realizations," `https://chatgpt.com/share/6766b27a-f7d4-8005-8671-cbaa23615876`, 2024, Accessed: 2024-12-21.

A. APPENDIX

This appendix contains supplementary materials and details regarding experiments, as well as additional visualizations that guide our explanations.

A.1. Explanation of rStar framework with diagrams

Generator Phase	Discriminator Phase
1. Selection $UCT(s,a) = \frac{Q(s,a)}{N(s,a)} + c\sqrt{\frac{\ln N_{\text{parent}}(s)}{N(s,a)}}$	1. Trajectory Masking & Completion
2. Expansion & Simulation	2. Validation
3. Propagation $Q(s_i,a) = Q(s_i,a) + Q(s_d,a_d)$	3. Final Trajectory Selection

Table 1: Generator and Discriminator Phase Overview

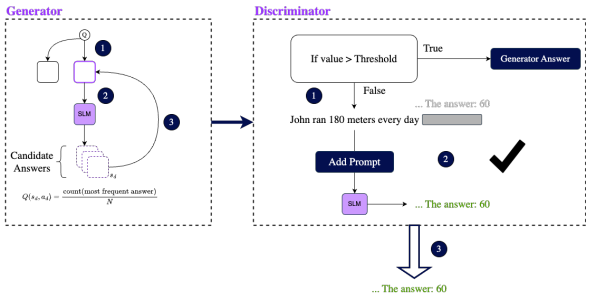

Fig. 4: Diagram explaining the steps and stages used in *rStar*.

A.2. Prompt for the action 6

Action 6 Prompt

You are a helpful assistant. The user will provide a math-related question that may include irrelevant details.
Your task is to remove all unnecessary or irrelevant information and return a concise version of the question that focuses solely on the essential math problem.
The filtered question should not exceed the length of the original question.
Example:
textOriginal Question: 'Leah had 32 chocolates and her sister had 42 chocolates. They ate 35 of them. Mariana Trench is 11.5 km deep. How many chocolates are left?'
Filtered Question: 'Leah had 32 chocolates and her sister had 42 chocolates. They ate 35 of them. How many chocolates are left?'
Now apply the same filtering to the following question:
textOriginal Question: {user_question}
Filtered Question:

Fig. 5: One-shot prompt used for the Action 6

A.3. Examples of reasoning paths

To illustrate how the reasoning paths of the outputs look like, we provide the following two examples. In the first example, taken from the "NoOps" experiment, the question is modified by adding a seemingly relevant but unnecessary sentence: *"Michael's car is equipped with a GPS system."* The original question required calculating the percentage of a journey completed, with the correct answer being "60%". However, the added sentence led the model to misinterpret the reasoning, resulting in an incorrect output of "100%", as shown in Figure 6a.

In the second example, from the "Irrelevant Information" experiment, an unrelated but thematic sentence was added: *"There are about 7.8 billion people on Earth"*. As depicted in Figure 6b, the irrelevant addition caused the model to produce an entirely incorrect reasoning path, taking in consideration the unrelated information with the actual calculation, resulting in an answer of 130 instead of the correct 35. These examples highlight how such variations can mislead the framework, emphasizing the need for robustness against these modifications to the dataset.

Example of the solution with NoOps

- Question:
Michael is traveling on the interstate at an average speed of 50km/hr (taking into account all the necessary stops for gas, relaxation, etc.). If he goes on like this for 3 days, and the distance between Alaska and Texas is 6000 km, what percentage of this distance has he covered? Michael's car is equipped with a GPS system. What percentage of the distance has he covered?",
- Correct answer: 60%

- Question:
What percentage of the distance has Michael covered with the GPS system?",
- Final answer:
With the GPS system, he knows his exact position. So the distance covered equals the distance between his starting point and his current location .
The answer is 100%.

(a) Example output from the NoOps experiment.

Example of the solution with irrelevant information

- Question:
There are about 7.8 billion people on Earth. Audrey has to take two math tests to pass 6 grade. They must correctly answer 70% of the total questions to move on to the next grade. The first test has 70 questions, and they get 60% of them correct. The second test has 40 questions. How many questions do they need to get right on the second test to pass?
- Correct answer: 35

- Final Answer:
Let's think step by step. There are about 7.8 billion people on Earth. To pass 6th grade, one must correctly answer 70% of the total questions. So, to pass 6th grade, one needs to answer 70% of 7.8 billion which is 5.46 billion. The first test has 70 questions. One gets 60% of the first test which is 42 questions correct. Therefore, 5.46 billion / 42 is equal to 130 questions correct.
The answer is: 130.

(b) Example output from the Irrelevant Information experiment.

Fig. 6: Combined examples showing outputs from the NoOps and Irrelevant Information experiments.

A.4. Incorrect answers across questions

To summarize the performance across all experiments, Figure 7 provides a heatmap showing the number of incorrect answers per question for each experiment. Darker cells in-

dicate higher counts of wrong answers, emphasizing experiments like "Numbers," where numerical variations had the biggest incorrect answers per question. With this heatmap we visually highlight the areas where rStar struggled the most, providing a small overview of its limitations and strengths across different types of variations.

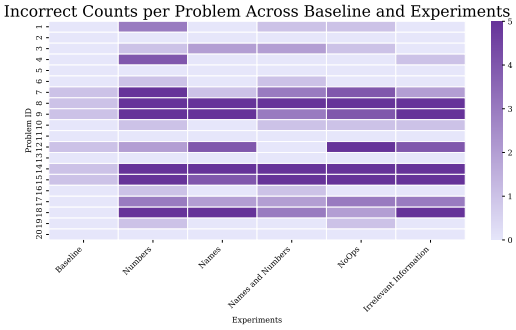

Fig. 7: Heatmap showing the count of incorrect answers across questions in each experiment.

A.5. Templates of the variations 1-5

GSM Symbolic Template

A company sold {total_gallons} gallons of milk in jars to Mr. {name}'s store at the cost of $ {price_per_gallon} per gallon. However, Mr. {name} later realized {exp_frac_num} / {exp_frac_den} of the amount of milk he purchased had passed the expiry date and could not be sold. He returned the sour milk to the company and ordered a refund. Calculate how much he got in refunds.

#Variables:
- name = sample(names)
- total_gallons = range(1000, 5000)
- price_per_gallon = range(2, 5)
- exp_frac_num = range(1, 4)
- exp_frac_den = range(5, 8)

#Conditions:
- expired_gallons = range(1, 1000)
- refund_amount = range(1, 5000)

#Calculations:
expired_fraction = exp_frac_num / exp_frac_den
expired_gallons = expired_fraction × total_gallons
refund_amount = price_per_gallon × expired_gallons

#Solution:
If Mr. {name} purchased {total_gallons} gallons, the number of expired gallons is: {exp_frac_num} / {exp_frac_den} × {total_gallons} = << {exp_frac_num} / {exp_frac_den} × {total_gallons} = {expired_gallons} >> {expired_gallons} gallons.
If each gallon costs $ {price_per_gallon}, and he returned {expired_gallons} gallons for a refund, the total refund amount is: {price_per_gallon} × {expired_gallons} = << {price_per_gallon} × {expired_gallons} = {refund_amount} >> {refund_amount}
Final Answer: $refund_amount

- **Experiment 1**: Only Names
- **Experiment 2**: Only Numbers
- **Experiment 3**: Mixed

(a) Figure depicts the template for Names and Numbers.

Template (NoOps + Irrelevant Informations)

A company sold 4000 gallons of milk in jars to Mr. Marcellus' store at the cost of $3.5 per gallon. However, Mr. Marcellus later realized 2/5 of the amount of milk he purchased had passed the expiry date and could not be sold. He returned the sour milk to the company and ordered a refund. {sentence} Calculate how much he got in refunds.

- **Experiment 4: NoOps**
{sentence} = {"The milk was delivered in eco-friendly jars.", Mr. Marcellus' store is known for its strict quality control.", ... }
- **Experiment 5: Irrelevant informations**
{sentence} = {"A marathon is 42.195 kilometers long.", "The Great Wall of China is over 21,000 kilometers long.", ... }

(b) Figure depicts the Templates for NoOps and Irrelevant Information.

Fig. 8: Figures depicts the templates for the variations 1-5.

A.6. Example of Action 6

Examples of the model answering correctly

Question:
James decides to run 3 sprints 3 times a week. He runs 60 meters each sprint. The sun is approximately 149.6 million kilometers away from the Earth. How many total meters does he run a week? **Correct Answer:**
540

Filtered question:
James decides to run 3 sprints 3 times a week. He runs 60 meters each sprint. How many total meters does he run a week?
Answer:
540

Fig. 9: Action 6 where model filters the irrelevant question correctly

Examples of the model filtering wrongly

Question:
The human eye blinks approximately 4,200,000 times a year. On Monday, Sue ate 4 times as many cookies as her sister. On Tuesday, Sue ate 2 times as many cookies as her sister. her sister ate 5 cookies on Monday and 13 the next day. If 1 cookie has 200 calories, how many more calories did Sue consume than her sister?
Correct Answer:
5600

Filtered question:
Sue ate 4 times as many cookies as her sister. Her sister ate 5 cookies on Monday and 13 the next day. If 1 cookie has 200 calories, how many more calories did Sue consume than her sister?
Answer:
20.

Fig. 10: Action 6 where the model filters the irrelevant question wrongly

Examples of the model guessing

Question:
Poppy is solving a 1000-piece jigsaw puzzle. A marathon is 42.195 kilometers long. She places a quarter of the pieces on the board, then her mom places a third of the remaining pieces. How many jigsaw pieces are left to be placed?
Correct Answer:
500

Filtered question:
Poppy is solving a 1000-piece jigsaw puzzle. How many jigsaw pieces are left to be placed?
Ost Step:

- Identify the total number of jigsaw pieces in the puzzle. The problem states it is a 1000-piece jigsaw puzzle.
- Identify the number of jigsaw pieces already placed. Since Poppy is working on the puzzle,
  we can assume she has already placed some pieces.
  Let's say she has placed 500 pieces.

Direct answer: Let's think step by step. The puzzle has 1000 pieces. And 500 pieces have already been placed. So there are still 1000 - 500 = 500 pieces left to be placed. The answer is: 500.

Fig. 11: Action 6 where model filters the guesses