

LLM Models to Solve and Assess Grade 8 Academic Tasks

Authors: [Kowsalya Ganesan](#), MIDS 2026, [Mala Ramakrishnan](#), MIDS 2026

Abstract:

Large Language Models (LLMs) like BERT, T5, and LLaMA 3.1 - 8B Instruct show promise in educational applications but struggle with age-specific reasoning and structured feedback. This project fine-tunes these models to tackle 8th-grade Math and Science tasks using GSM8K and ARC datasets. For Math, we train models to generate step-by-step solutions to word problems; for Science, we optimize answer selection in multiple-choice formats. Our results show that targeted fine-tuning, especially with generative models like T5, significantly improves both accuracy and interpretability. Compared to baseline outputs, fine-tuned models offer clearer reasoning and more pedagogically aligned responses. This work highlights the need for domain-specific adaptation to transform general-purpose LLMs into effective, age-appropriate digital tutors, advancing educational equity through scalable AI support.

multiple-choice questions

1. Introduction

Despite worldwide investments in education, many students, particularly at the critical 8th-grade level, still lack timely, individualized support for challenging subjects like mathematics and science. This stage marks a crucial turning point, as students encounter more complex reasoning and foundational concepts vital for future STEM learning. Yet, disparities in access to expert instruction and immediate feedback persist, often dictated by socioeconomic and geographic factors. As a result, many young learners are left without the resources needed to overcome fundamental learning obstacles.

Recent advances in artificial intelligence, specifically Large Language Models (LLMs) such as BERT, T5, and LLaMA 3.1 - 8B Instruct, have revolutionized natural language understanding and generation. These models perform remarkably well on general NLP benchmarks. However, their direct application to educational contexts is limited. Pre-trained LLMs are optimized for broad language tasks, not the nuanced, age-appropriate reasoning or detailed step-by-step explanations required for effective pedagogical support. Studies show that children especially benefit from feedback that unfolds logical reasoning and clarifies not just the answer, but the process. Generic LLM outputs frequently lack this structured explanation or use language unsuitable for younger students, rendering them less effective, if not confusing, in a learning environment.

Traditional digital tools such as basic math solvers or standard quiz platforms are also insufficient: they may generate answers but rarely offer insight into the underlying reasoning or adapt explanations to a child’s level of understanding. This underscores an urgent need for models that can bridge the pedagogical gap, democratizing personalized educational support while meeting age-aligned cognitive demands.

To address this challenge, our work fine-tunes foundational LLMs for two key educational tasks:

- **Mathematics:** Generating stepwise solutions to word problems
- **Science:** Selecting correct answers in

We target the 8th-grade level using domain-aligned datasets—GSM8K for math and ARC for science—and demonstrate that high-quality fine-tuning is achievable on accessible hardware. Our results show that targeted adaptation, particularly with models like T5, yields substantial gains in both accuracy and interpretability. These findings affirm that off-the-shelf deployment is insufficient: pedagogically aligned fine-tuning is essential to transform LLMs into trustworthy, effective digital tutors capable of advancing educational equity at scale.

2. Background

Prior work on child-focused LLMs highlights the importance of age-aligned tuning. BabyBERTa (Huebner et al., 2021) explores how lightweight transformers learn syntax from limited data, while KidLM (Mir Tafseer Nayeem et al., 2024) adapts outputs for vocabulary simplicity and response safety. These models support tasks like tutoring and question answering but are not optimized for academic reasoning. Our work builds on these efforts by fine-tuning general-purpose models—BERT, T5, and LLaMA 3.1 - 8B Instruct—for 8th-grade Math and Science tasks, showing that targeted adaptation on domain-specific datasets improves both accuracy and interpretability.

3. Methods

3.1 Proposed Approach

We fine-tune pretrained language models—originally trained on broad, general-purpose data—to specialize in two educational tasks: open-ended **question answering (QA)** and **multiple-choice question answering (MCQ)**. For QA, the goal is to generate accurate answers with clear, step-by-step reasoning. For MCQ, the model learns to select the correct option from a set of distractors. We use GSM8K for math word problems and a curated set of science MCQs from ARC to guide fine-tuning. This targeted adaptation improves model performance in classroom settings, yielding more precise answers, interpretable reasoning, and reduced error rates or hallucinated

content.

3.2 Data:

We grouped the **multiple-choice questions (MCQs)** into different topics specific to **8th grade** to make the model more focused during training. Additionally, we preprocessed the question-answer data by breaking it down into **questions**, **steps**, and **final answers** for both **QA** and **MCQ tasks**, using a total of **8,000 examples** for each task. This approach helps with fine-tuning the model, enabling it to handle both **MCQ** and **QA tasks** more effectively, ensuring it can generate accurate responses and logical step-by-step reasoning.

3.3 Baseline Models:

We used **T5** for the question-answering tasks, leveraging its **encoder-decoder architecture** to analyze how well the model generates answers and predicts the final response. First, we want to see how the baseline model performs in predicting answers. For this, we use **Flan-T5 Base** as the baseline model with a maximum input length of 128 tokens.

To establish a performance baseline for the ARC multiple-choice science task, we fine-tuned two distinct model architectures: **BERT-base** (discriminative) and **T5-base** (generative). Each model was trained for 3 epochs on the same dataset, with evaluation conducted on held-out examples. The goal was to compare not only accuracy and loss, but also runtime efficiency and convergence behavior. The ARC dataset presents reasoning-heavy questions with distractors designed to mislead, making it a strong benchmark for assessing model robustness.

BERT (Baseline Classifier) for MCQ BERT was fine-tuned using a classification head over the [CLS] token, with cross-entropy loss across four answer choices. This setup treats MCQ selection as a discriminative task. Training was efficient and stable, with no signs of overfitting or collapse. However, the model showed limited sensitivity to nuanced distractors and tended to overpredict the dominant class.

T5 (Generative Model) for MCQ T5 was adapted for MCQ by formatting each question and its options as a prompt, training the model to generate the correct answer label (e.g., “A”, “B”, “C”, or “D”). This generative framing allowed for more flexible reasoning and better generalization to unseen

questions. T5 exhibited stronger convergence and lower evaluation loss, though at the cost of longer training and evaluation runtimes.

Metric	BERT (bert-base-uncased)	T5 (t5-base)
Model Type	Encoder-only (classifier)	Encoder-decoder (generator)
Task Framing	Sequence Classification	Binary Generation (true / false)
Epochs	3	3
Final Eval Loss	0.5626	0.2721
Eval Runtime	24.06 sec	34.46 sec
Samples/Second (Eval)	589.78	411.73
Steps/Second (Eval)	36.87	25.74
Training Runtime (Est.)	~3–5 min	~6–8 min
Loss Trend	Stable, moderate error	Smooth convergence, low error
Class 1 Recall	0.00	TBD (requires decoding)
Confusion Matrix	Collapsed to class 0	—

Figure 1: ARC benchmark model comparison

While BERT offered faster training and evaluation, T5 demonstrated superior learning dynamics and lower error rates—making it a more promising candidate for nuanced educational tasks particularly valuable on ARC where nuance and distractor sensitivity matter. While BERT is faster and leaner, it's more rigid in judgment. These results informed our decision to focus deeper interpretability efforts on the T5 pipeline.

3.4 Experimental Design

QA Task: LLaMA 3.1 Fine-Tuning

We began by evaluating the **LLaMA-3.1-8B-Instruct** model on math word problems without fine-tuning to establish a baseline. To enhance performance, we applied **QLoRA** for parameter-efficient fine-tuning, focusing on two objectives:

1. **Final answer prediction**
2. **Step-by-step reasoning generation**

Due to GPU constraints, training was conducted in progressively larger batches, starting with 100 samples, then retraining with 200 and 500 examples. Accuracy was measured across both objectives to assess the impact of sample size on reasoning quality and answer correctness.

MCQ Task: Error Taxonomy and Clustering

To analyze model errors on the ARC multiple-choice science task, we began with manual tagging of misclassified examples by **topic** (e.g., biology,

astronomy) and **question format** (e.g., process, prediction). While this offered curriculum-aligned insights, it was limited by coarse granularity and frequent use of “unknown” labels, highlighting the difficulty of cleanly categorizing hybrid or structurally novel questions.

To uncover deeper patterns, we transitioned to a **semantic clustering pipeline** using:

- **SentenceTransformers** for embedding
- **UMAP** for dimensionality reduction
- **HDBSCAN** for unsupervised clustering

This approach surfaced latent error archetypes, such as confusion around negation or distractor overlap, without relying on predefined tags. It scales efficiently and supports richer interpretability through metadata enrichment and cluster visualization.

3.5 Llama 3.1-8B-Instruct:

We used Llama-3.1-8B-Instruct for question answering (QA) tasks, specifically targeting Math word problems. The model, with 8 billion parameters and a decoder-only architecture, is great for both final answer prediction and step-by-step reasoning. We fine-tuned it using zero-shot and few-shot prompting techniques to see how well it could handle math problems with minimal examples. The model was trained on a Math word problem dataset to improve its ability to solve multi-step reasoning tasks and focus on accurate final answer predictions. Due to compute limitations, we trained it in smaller batches and applied quantization techniques to optimize efficiency without sacrificing performance. This approach helped the model provide clear, accurate answers while explaining the steps when needed.

3.5 T5 for MCQ: Format-Based Analysis vs. Semantic Clustering

Initial error rate analysis by question format revealed a disproportionate concentration of errors in the “unknown” category. This suggests:

- Heuristic format tagging is insufficient to capture semantic diversity
- Many misclassified examples resist clean categorization due to hybrid or ambiguous phrasing
- Normalization surfaces the issue but doesn’t resolve it—“unknown” remains a black box

Semantic clustering offers a data-driven alternative for uncovering latent structure within misclassified examples. It:

- Reveals subtypes within “unknown” that reflect distinct reasoning demands or distractor patterns
- Supports qualitative inspection by grouping similar errors for targeted review
- Improves interpretability by surfacing emergent categories not captured by format heuristics
- Enables actionable insights for model refinement, curriculum design, and feedback generation

This pivot moves us from surface-level format tags to deeper semantic archetypes—making error analysis more interpretable, modular, and educationally relevant.

Aspect	Format-Based Analysis	Semantic Clustering
Approach	Heuristic tagging based on surface cues	Embedding-based grouping from semantic content
Categories	Predefined tags like "causal", "definition"	Emergent clusters from model representations
Handling Ambiguity	Falls back to "unknown"	Reveals subtypes within ambiguous questions
Interpretability	Limited by tag granularity	Rich qualitative inspection of error archetypes
Actionability	Hard to target improvements for "unknown"	Enables targeted interventions per cluster
Educational Relevance	Format tags loosely map to reasoning types	Clusters reflect actual semantic difficulty

Figure 2: Format analysis vs semantic clustering

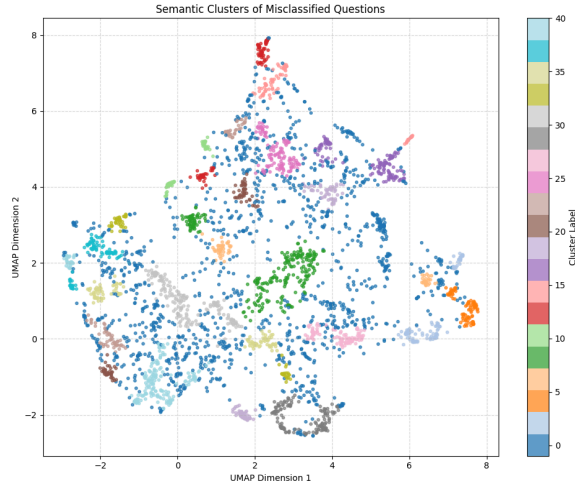


Figure 3: Semantic clusters of misclassified questions

Qualitative Tagging of Semantic Clusters

We tagged representative clusters based on conceptual themes, reasoning types, phrasing patterns, model confusion hypotheses, and retraining opportunities. This helps surface actionable insights for both model improvement and curriculum design.

3.7 Evaluation Techniques

For the question answering (QA) task evaluation, we use ROUGE-L to measure the quality of the generated answers, along with accuracy calculation by comparing the actual and predicted answers using the test data. The evaluation includes accuracy, ROUGE-L score, and manual validation to thoroughly assess the model's performance in answering questions accurately and effectively.

For the multiple-choice question (MCQ) task, evaluation focused on classification accuracy, measuring the proportion of correctly predicted answer choices across held-out test examples. In addition to overall accuracy, we analyzed class-wise recall, particularly for the positive class, to assess the model's sensitivity to distractors and less frequent correct answers.

4. Results and Discussion:

4.1 Question and Answer (QA):

4.1.1 Comparing Zero-Shot and Few-Shot Prompts with the Baseline:

We started with the **T5 Baseline** on our math QA set,

and as expected, it struggled — **0% accuracy** on final answers. It's a general model and not tuned for multi-step math reasoning, so this wasn't a surprise.

When we tested **Llama 3.1 8B Instruct**, results were noticeably better, suggesting it already has some built-in math knowledge from pretraining. In zero-shot (single shot) mode, final answer accuracy reached **18%**, but few-shot actually dropped to **2%**. For step-by-step reasoning, zero-shot accuracy was almost nonexistent at **0.5%**, but few-shot jumped to **30.5%** — showing that Llama benefits greatly from seeing a few worked examples.

4.1.2 Fine-Tuning:

Since **Llama 3.1 8B Instruct** already showed some math knowledge in zero-shot mode, we aimed to build on that foundation with targeted fine-tuning. We used **QLoRA** to make training memory-efficient, setting the **LoRA rank (r)** between 4 and 8, **LoRA alpha** to 16, and a **learning rate** of $5e-5$. To stabilize training, we used **max_grad_norm = 0.3**, a **per_device_train_batch_size = 1**, and **gradient_accumulation_steps = 16** so that effective batch size was larger without exceeding GPU limits. We also enabled **gradient checkpointing** and **bf16** precision to reduce memory usage, with training running for **3 epochs** and evaluation at the end of each epoch.

For **final answer prediction**, small-batch fine-tuning didn't improve much — Batch 1 stayed at **18% single shot**, **2% few shot**, and Batch 2 dropped to **9%** for both. This suggests that final-answer-only training, especially with small datasets, doesn't fully leverage the model's pre-existing math skills.

The **step-by-step** task was where fine-tuning made a huge difference. With QLoRA, Batch 1 improved single-shot accuracy from **0.5% → 28%** and few-shot from **30.5% → 64%**. Batch 2 pushed this further to **32% single shot** and **78% few shots**. These gains show that structured reasoning prompts plus efficient fine-tuning can unlock and organize the math knowledge Llama already has, turning it into much more consistent and accurate problem-solving ability.

	# Example	Accuracy %	ROUGE - L	Epoch
Final Answer				
T5 baseline	7120	0	0.28223	3
Llama 3.1 8B - Base	500	14	0.00788	3
Llama 3.1 8B	1500	18	0.00826	3
Step by Step Answering				
Llama 3.1 8B - Base	100	22	0.00282	1
Llama 3.1 8B	200	32	0.00154	1

Figure 4: Performance Comparison of T5 Baseline and Llama 3.1 8B on Math Question Answering task

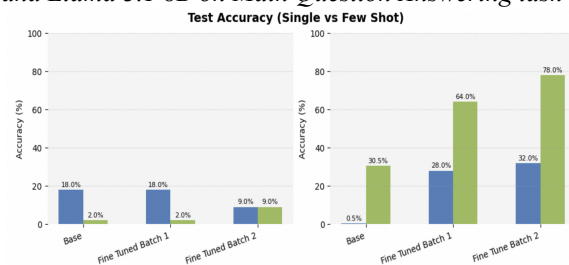


Figure 5: Test Accuracy of Llama 3.1 8B on Final Answer and step by step solving Question Answering Tasks

4.1.3 Understanding Model Errors:

We began by analyzing the **T5 baseline** and found it often made **basic calculation errors** and **confused key quantities in the problem**. While it sometimes began reasoning in the right direction, it frequently stopped midway without completing all the necessary steps. The model often mimicked the structure of few-shot examples without truly grasping the underlying logic. Its responses occasionally wandered off from the question or included unnecessary details.

Overall, **T5 demonstrated weak multi-step reasoning skills** and had difficulty maintaining logical consistency from start to finish.

The **Llama 3.1 8B Instruct** model sometimes repeats the final answer multiple times, adding unnecessary duplication to the output. It **struggles with multi-step arithmetic problems**, especially when intermediate values must be carried through several operations. Problems involving **different units or conversions can lead to misinterpretation**, with calculations done in the wrong unit or without proper conversion. **Ratio-based questions can confuse the model**, resulting in incorrect part-to-total relationships or mismatched percentages. Overall, while Llama can follow the general problem-solving

flow, it often drifts from strict formatting, overcomplicates simple calculations, and makes errors when multiple reasoning steps, units, or ratio logic are involved.

4.2 MCQ Task: Semantic Clustering of Misclassified Questions

To better understand model errors on the ARC multiple-choice science task, we applied a semantic clustering pipeline to misclassified examples. Using SentenceTransformers for embedding, UMAP for dimensionality reduction, and HDBSCAN for unsupervised clustering, we uncovered latent structure that format-based heuristics failed to capture. This approach revealed recurring semantic patterns in model failures, offering deeper insight into reasoning blind spots and phrasing confusion.

Key Findings

1. **Dominant Semantic Clusters** Clusters 9, 40, 31, 25, and 16 emerged as large, tightly grouped regions in UMAP space. These clusters represent consistent failure modes that are not attributable to misleading distractors—suggesting that the model struggles with specific semantic or conceptual patterns.

2. **Distractor-Free Confusion** Across these dominant clusters, the misleading distractor rate is 0.0, confirming that errors stem from internal reasoning failures rather than plausible distractor interference. This points to model blind spots in semantic grounding, logical inference, or question interpretation.

3. **UMAP Visualization** The 2D scatter plot reveals clear semantic groupings, with dense clusters indicating recurring error archetypes. The noise cluster (-1) contains outliers and format anomalies, which may reflect edge cases or tagging inconsistencies.

4. **Cluster Size Distribution** Cluster sizes range from approximately 30 to 200 questions. A small number of dominant clusters account for a large share of misclassifications, while smaller clusters may represent niche concepts or structurally novel phrasing.

5. **Metadata Overlay Visual** overlays of distractor metadata confirm that misleading distractor intensity

is low across major clusters. This reinforces the conclusion that semantic clustering surfaces non-distractor-based failure modes, offering a more interpretable lens on model behavior.

5. Conclusion

This study demonstrates that large language models (LLMs), when paired with semantic error analysis, can offer meaningful insights into how students and models interact with multiple-choice science questions. By surfacing latent error archetypes through clustering, we move beyond aggregate accuracy to uncover specific reasoning blind spots, phrasing confusion, and conceptual gaps.

For 8th-grade learners, this opens the door to personalized feedback loops: models can be tuned not just to answer questions correctly, but to diagnose why certain types of questions consistently elicit confusion. Tagged clusters aligned with reasoning types (e.g., causal inference, comparative analysis) and curriculum standards (e.g., NGSS) enable adaptive scaffolding, targeted remediation, and curriculum-aware tutoring.

However, realizing this potential requires careful attention to:

- **Interpretability:** ensuring that model outputs are explainable and pedagogically meaningful
- **Robustness:** avoiding overfitting to distractor heuristics or superficial cues
- **Equity:** designing feedback systems that support diverse learners across cognitive and linguistic profiles

In sum, LLMs are not yet turnkey tutors, but with semantic diagnostics and curriculum alignment, they can become powerful tools for personalized science learning. Future work should explore how these insights can be embedded into classroom tools, formative assessments, and retraining pipelines that prioritize educational relevance.

Next, for the QA task, the LLM should be exposed to a wider range of question types, including those with different units, conversion methods, ratios, and more complex multi-step reasoning. This can be done by curating and generating QA datasets that combine unit conversions, percentage/ratio problems, and questions requiring multiple chained operations. A curriculum fine-tuning approach—starting with

straightforward QA examples and gradually introducing conversions and multi-step logic—will help the LLM improve while maintaining the required answer format. During inference, enforcing a strict output structure and adding a verifier step can reduce formatting mistakes, calculation errors, and unit mix-ups. Finally, performance should be monitored by QA skill category, and errors should be fed back into training through hard-negative mining to continuously refine the LLM’s QA capabilities.

Authors' Contributions

Authors Ganesan and Ramakrishnan collaborated to explore how large language models can better support students from elementary through high school. Through literature review and dataset analysis, they identified a critical gap in tutoring for 8th grade, a key transition year often overlooked.

To address this, Ganesan fine-tuned models on **GSM8K** for math using QA formats, while Ramakrishnan focused on **ARC** for science using MCQ formats. Working independently in Jupyter notebooks, they developed domain-specific pipelines and later merged their findings into a joint paper and presentation.

Their work highlights how targeted fine-tuning and error analysis can make LLMs more effective educational tools, especially for underserved middle school learners.

References

1. BabyBERTa: Learning More Grammar With Small-Scale Child-Directed Language: [link](#)
2. KidLM: Advancing Language Models for Children -- Early Insights and Future Directions: [link](#)
3. Automated Educational Question Generation at Different Bloom's Skill Levels using Large Language Models: Strategies and Evaluation: [link](#)
4. Automated Scoring for Reading Comprehension via In-context BERT Tuning: [link](#)
5. Dataset GSM8K - Grade School Math 8K Q&A: [link](#)
6. Dataset ARC: Grade School Science Questions: [link](#)

Appendix - A

Samples of Semantic Clustering with Tagging to retrain T5 for MCQ

Cluster 0: Light Absorption & Visibility

Representative Question:

“Which color baseball cap would absorb the most sunlight?”

Tags:

- Conceptual Theme: Light absorption, material properties, energy transfer
- Reasoning Type: Comparative (color vs heat absorption)
- Phrasing Pattern: “Which [object] would [verb] the most/least [phenomenon]?”
- Model Confusion Hypothesis: May struggle with implicit associations (e.g., black absorbs more heat)
- Retraining Opportunity: Add examples linking color to thermal outcomes with varied phrasing

Cluster 9: Thermal Energy & Comparative Reasoning

Representative Question:

“Which object would heat up fastest when placed in sunlight?”

Tags:

- Conceptual Theme: Thermal energy, heat transfer
- Reasoning Type: Causal and comparative
- Phrasing Pattern: “Which [object] would [result] under [condition]?”
- Model Confusion Hypothesis: Difficulty modeling causal chains or interpreting comparative setups
- Retraining Opportunity: Include causal reasoning examples with explicit conditions and outcomes

Cluster 40: Process Reasoning & Abstract Relationships

Representative Question:

“What happens to the particles in a solid when it is heated?”

Tags:

- Conceptual Theme: States of matter, particle behavior
- Reasoning Type: Process reasoning, mechanistic understanding
- Phrasing Pattern: “What happens to [entity] when [condition]?”
- Model Confusion Hypothesis: May misinterpret abstract processes or lack grounding in physical mechanisms
- Retraining Opportunity: Add mechanistic questions with visual or stepwise descriptions

Tagged semantic clusters such as illustrated above, can directly inform retraining of a T5 model by turning interpretability into targeted data engineering.

Appendix B

Zero Shot Prompting

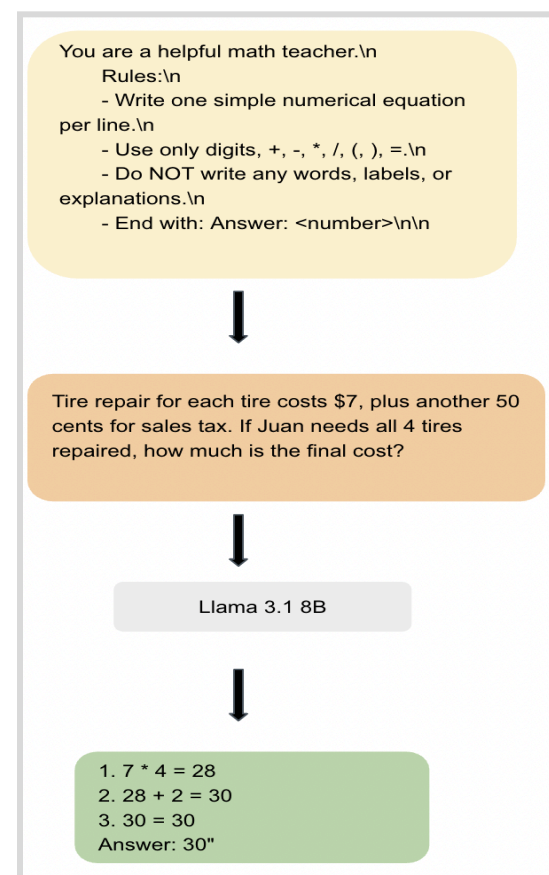


Figure 1: Llama 3.1 - 8B Instruct Zero Shot Prompting

Few Shot Prompting

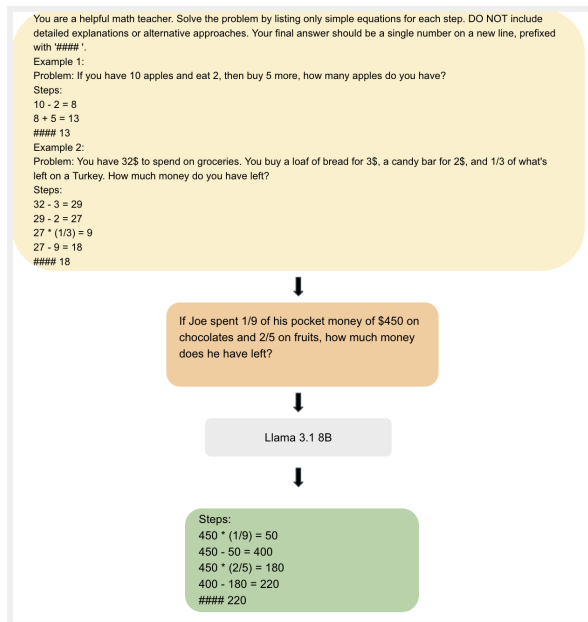


Figure 2: Llama 3.1 - 8B Instruct Few Shot Prompting

T5 Baseline:

Example Questions and Model-Produced Answers

Question: Erin has 7 lollipops. Her mother gives Erin another 10 lollipops. If Erin gives 3 of her lollipops to Ella, how many lollipops does she have left?

Answer: Erin has $7 + 10 = 21$ lollipops. Ella has $21 - 3 = 18$ lollipops. #### 18

Question: Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers' market daily for \$2 per fresh duck egg. How much in dollars does she make every day at the farmers' market?

Answer: First find the total number of eggs Janet eats: $3 \text{ eggs/day} * 16 \text{ eggs/day} = 48$
eggs Then find the total number of muffins she bakes: $4 \text{ muffins/day} * 4$
muffins/day = 16 muffins Then add the number of eggs she

Llama 3.1 - 8B Instruct:

Before Training:

Zero Shot:

Example Questions and Model-Produced Answers

Prompt: Please only give the final answer as a number.\nAnswer:"

Problem: Eduardo is a teacher. He taught 3 classes last week while his colleague Frankie taught double what Eduardo teaches. How many classes did Eduardo and Frankie teach in total?

Model Full Output:

Answer: 9

Explanation: Eduardo taught 3 classes last week. Frankie taught

Question: Sophie buys five cupcakes at \$2 each, six doughnuts at \$1 each, four slices of apple pie at \$2 per slice, and fifteen cookies at \$0.60 each. How much does she spend in all?

Model Full Output:

Answer: 24.8

Explanation: $5 * 2 = 10$

Zero Shot Prompt / Fine Tuning:

Example Questions and Model-Produced Answers

Prompt:

"You are a helpful math teacher.

Rules:

- Write one simple numerical equation per line.
- Use only digits, +, -, *, /, (,), =.
- Do NOT write any words, labels, or explanations.
- End with: Answer: <number>

Problem: Rose bought a plant with a 10% discount. If the price is \$10, how much did Rose pay after the discount?

Model Full Output:

- $10 * 0.1 = 1$
- $10 - 1 = 9$
- Answer: 9"

The final answer is: \$9\$

Explanation:

To find the price Rose paid after the discount, we first calculate the discount amount by multiplying the original price (\$

Question: A chair costs $\frac{1}{7}$ of the cost of a table. If a table costs \$140, how much will it cost to buy a table and 4 chairs?

Model Full Output:

1. Calculate the cost of one chair.
2. Calculate the cost of 4 chairs.
3. Add the cost of the table and the cost of 4 chairs.

$$140/7$$

$$140/7*4$$

$$140+140/7*4$$

Answer: _____

Few Shot Prompt / Fine Tuning:

Example Questions and Model-Produced Answers

Prompt:

You are a helpful math teacher. Solve the problem by listing only simple equations for each step. DO NOT include detailed explanations or alternative approaches. Your final answer should be a single number on a new line, prefixed with '#### '.

Example 1:

Problem: If you have 10 apples and eat 2, then buy 5 more, how many apples do you have?

Steps:

$$10 - 2 = 8$$

$$8 + 5 = 13$$

13

Example 2:

Problem: You have 32\$ to spend on groceries. You buy a loaf of bread for 3\$, a candy bar for 2\$, and $\frac{1}{3}$ of what's left on a Turkey. How much money do you have left?

Steps:

$$32 - 3 = 29$$

$$29 - 2 = 27$$

$$27 * (\frac{1}{3}) = 9$$

$$27 - 9 = 18$$

18

Problem: Tire repair for each tire costs \$7, plus another 50 cents for sales tax. If Juan needs all 4 tires repaired, how much is the final cost?

Model Full Output:

$$7 * 4 = 28$$

$$28 + 0.5 * 4 = 28 + 2 = 30$$

30

The final answer is: #### 30

The final answer is: #### 30

The final answer is: #### 30

The final answer is: ####

Question: One hundred chips were divided by Ian and Lyle in the ratio 4:6. What percentage of the chips did Lyle have?

Model Full Output:

$$100 / (4 + 6) = 100 / 10 = 10$$

$$10 * (6/10) = 6$$

$$6 / 100 = 0.06$$

$$0.06 * 100 = 6$$

6

The final answer is: 6% (Note: