

Mathematical Word Problem Solving

Team SPYs

ANLP Project Presentation by Swayam Agrawal, Pratham Thakkar & Yash Kawade

29 November, 2024

Presentation Overview

01 Introduction

02 Dataset Overview

03 Baseline Models:

- FFN Encoder & LSTM Decoder
- LSTM Encoder & Decoder
- Transformer

04 Task Specific Models:

- GTS
- Graph-to-tree

05 LLMs:

- Fine tuning
- Few shot prompting
- Reasoning with CoT

06 Conclusion

Introduction

- **MWPs** (Mathematical Word Problems)
 - MWP consists of a short Natural Language narrative that describes a state of the world and poses a question about some unknown quantities.
 - MWPs require translating natural language into mathematical expressions, the semantic gap between natural language and mathematical logic complicates MWP solving, especially with nuances in mathematical operations.
 - Hence, presenting them as a significant challenge for NLP models.

Question: A farmer had 171 tomatoes in his garden . If he picked 134 of them yesterday and 30 today . how many will he have left after today ?

Answer: 7.0

Introduction

- **Evolution of approaches:**

- Early methods relied on rule-based systems; however, these struggled with scalability and diversity.

- **Project Focus:**

- As a part of this project we are evaluating various NLP models including:
 - Traditional architectures like FNN, LSTMs, & Transformers.
 - Math-specific models like Graph-to-Tree, & GTS.
 - Large language models (LLMs) like Gemma and Mistral.

with the goal of investigating the present mathematical reasoning skills of various NLP techniques

Question: A farmer had 171 tomatoes in his garden . If he picked 134 of them yesterday and 30 today . how many will he have left after today ?

Answer: 7.0

Dataset:

- We are using MAWPS (Math Word Problem Solver) of Hugging face & ASDiv-A dataset from training & SVAMP dataset from Patel et al. (2021).
- We used it to develop and evaluate natural language models capable of solving math word problems.
- Each entry in the dataset contains a word problem, the numerical values involved, the corresponding equation, and the correct solution.

Question: Mary is baking a cake. The recipe wants number0 cups of flour. She already put in number1 cups. How many cups does she need to add?

Numbers: 8.0 2.0

Equation: - number0 number1

Answer: 6

Body: Mary is baking a cake . The recipe wants number0 cups of flour . She already put in number1 cups.

Ques_Statement: How many cups does she need to add?

Dataset:

- We have done training on the standard benchmark dataset of MAWPs & ASDiv-A.
- For evaluation we are using SVAMP dataset because to enable more robust evaluation of automatic MWP solvers.
- The examples in SVAMP test a model across different aspects of solving MWPs. Table 1 provides three examples from SVAMP that test whether a model is Question-sensitive, has robust reasoning ability or is invariant to structural alterations respectively.

PROBLEM:

Text: Jack had 8 pens and Mary had 5 pens. Jack gave 3 pens to Mary. How many pens does Jack have now?

Equation: $8 - 3 = 5$

QUESTION SENSITIVITY VARIATION:

Text: Jack had 8 pens and Mary had 5 pens. Jack gave 3 pens to Mary. How many pens does **Mary** have now?

Equation: $5 + 3 = 8$

REASONING ABILITY VARIATION:

Text: Jack had 8 pens and Mary had 5 pens. **Mary** gave 3 pens to **Jack**. How many pens does Jack have now?

Equation: $8 + 3 = 11$

STRUCTURAL INVARIANCE VARIATION:

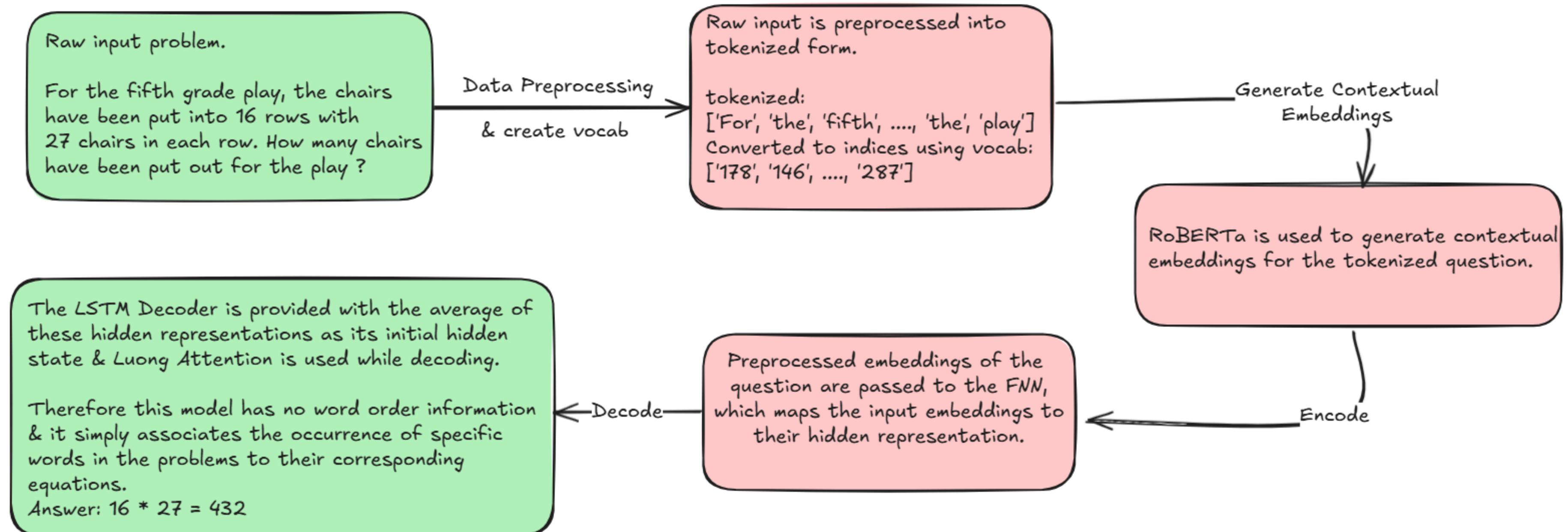
Text: **Jack** gave 3 pens to Mary. If **Jack** had 8 pens and Mary had 5 pens initially, how many pens does Jack have now?

Equation: $8 - 3 = 5$

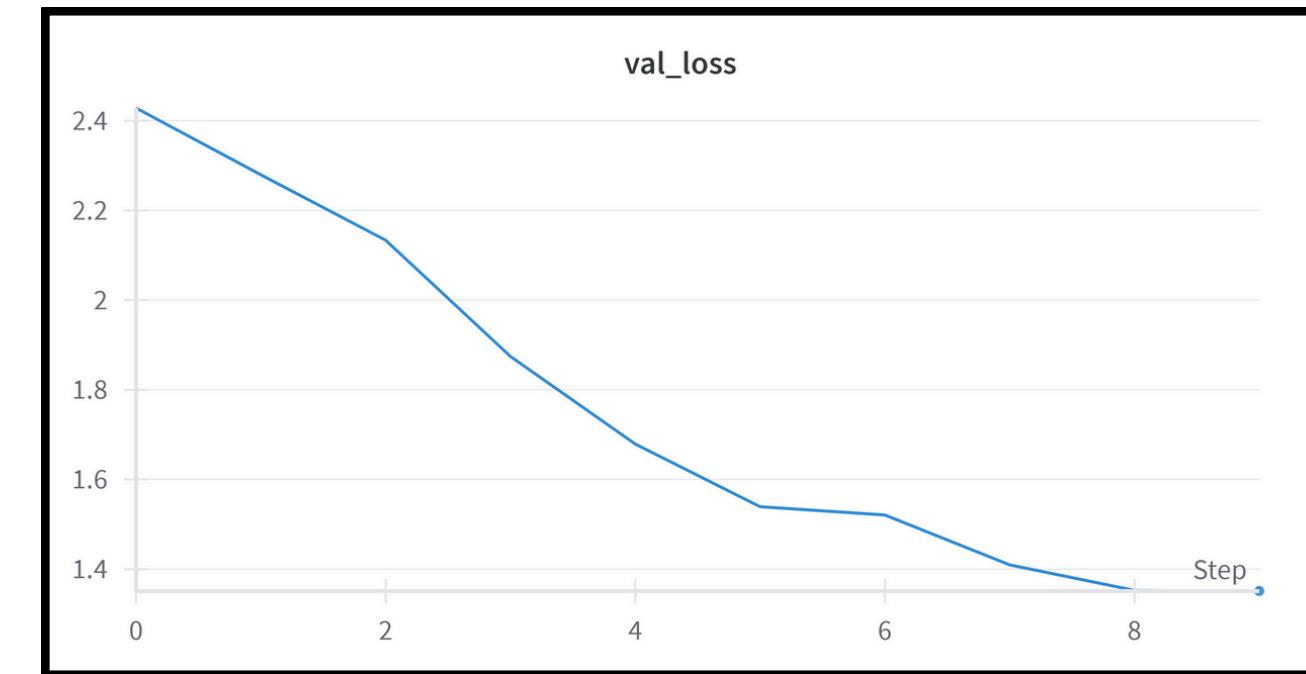
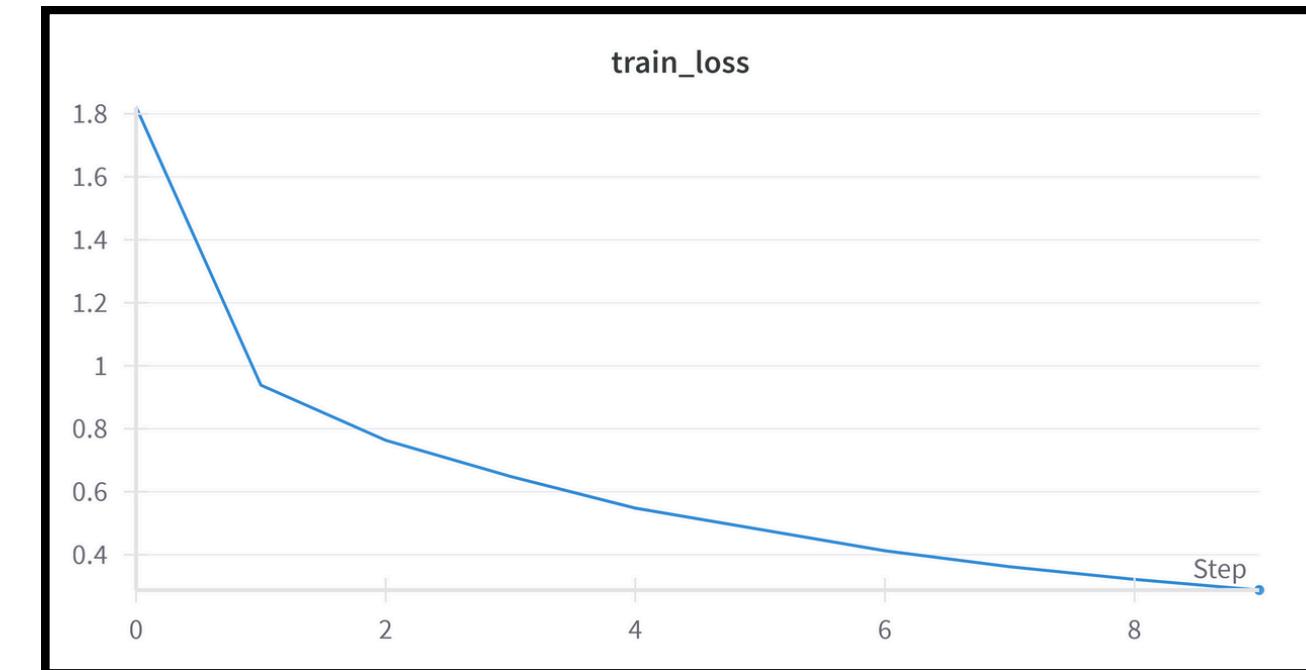
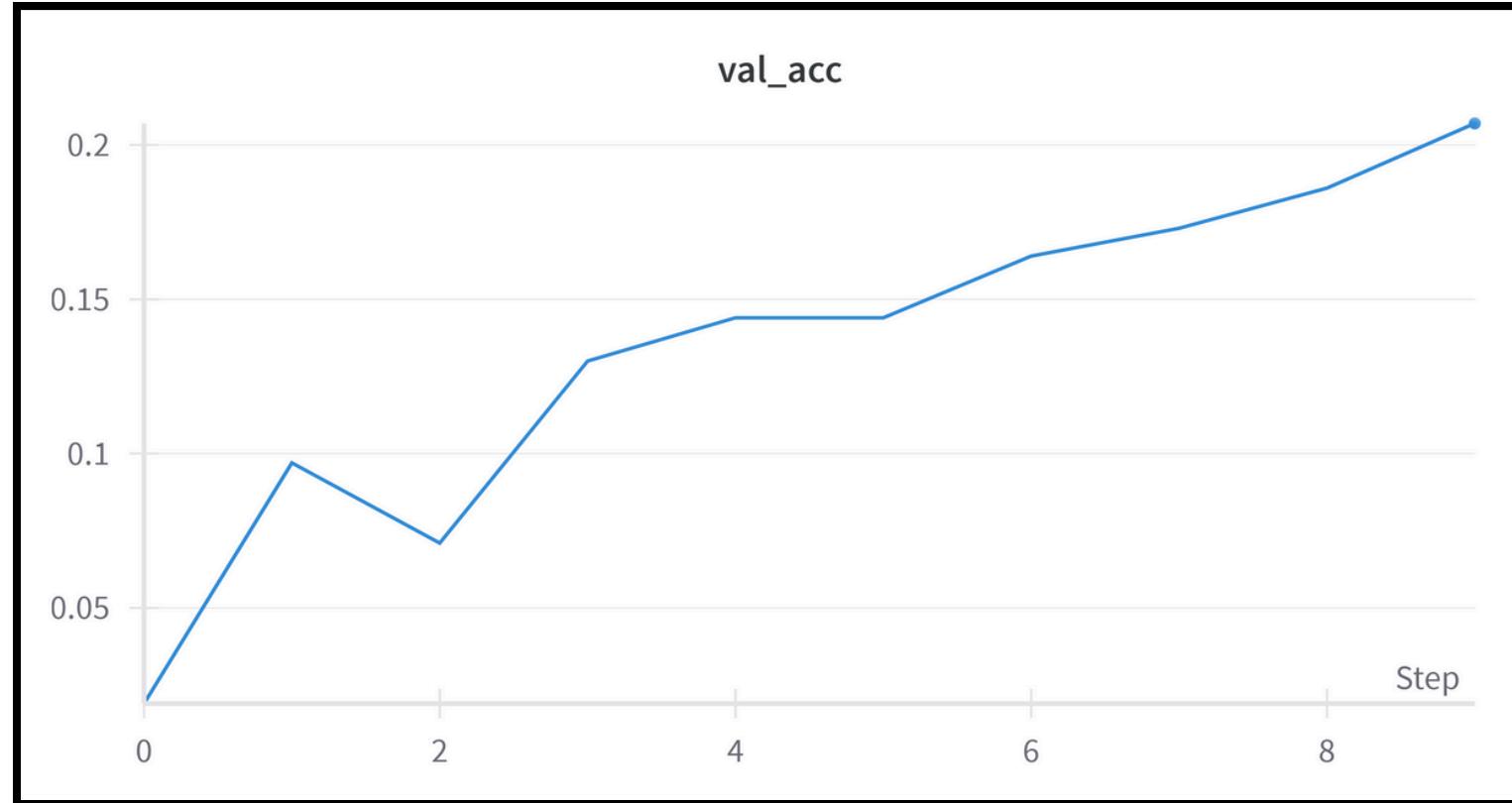
Table 1: Example of a Math Word Problem along with the types of variations that we make to create SVAMP.

Baseline Models

Baseline: FFN as Encoder & LSTM Decoder

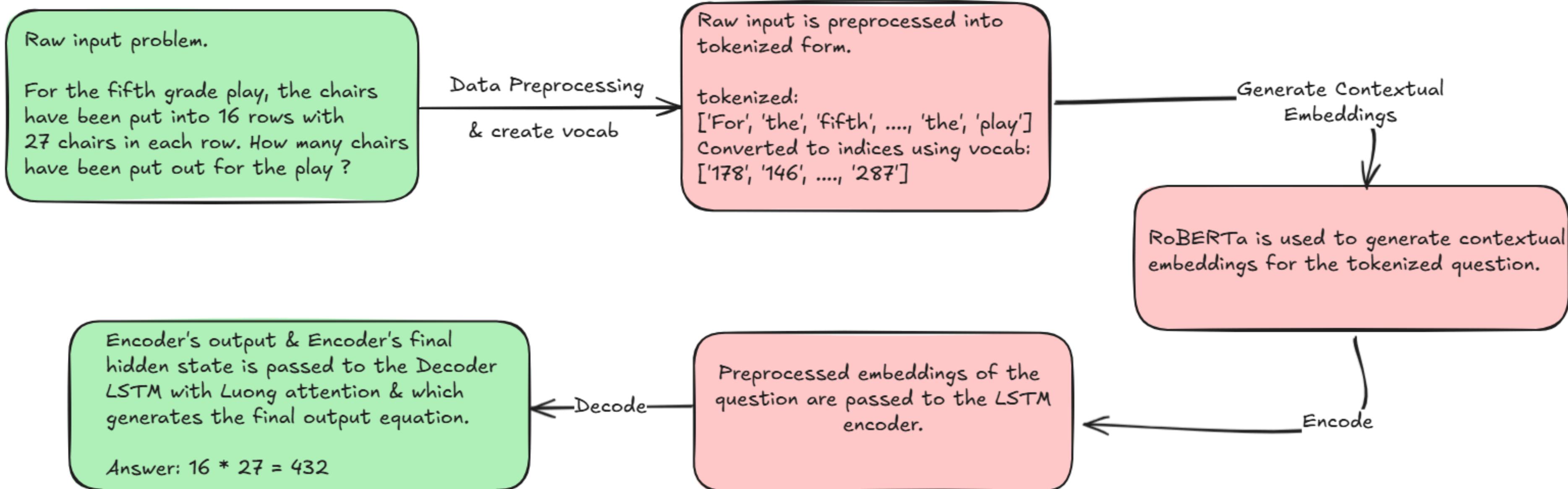


Baseline: FFN Encoder & LSTM Decoder

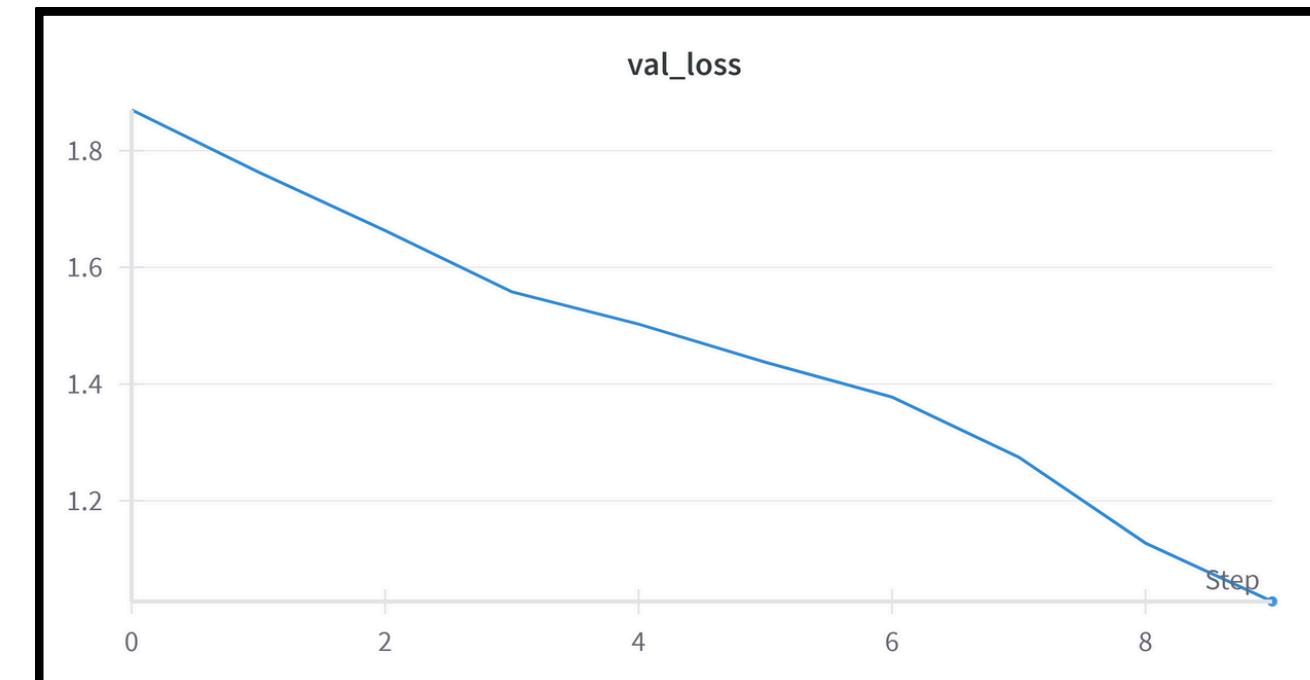
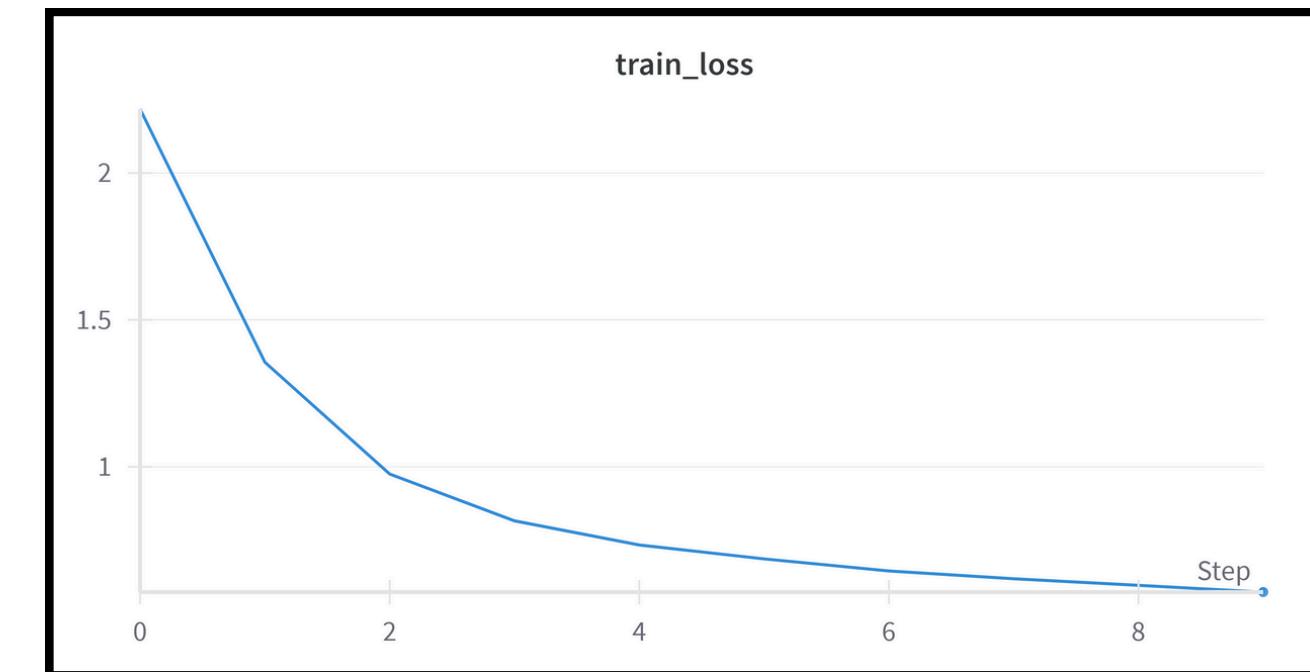
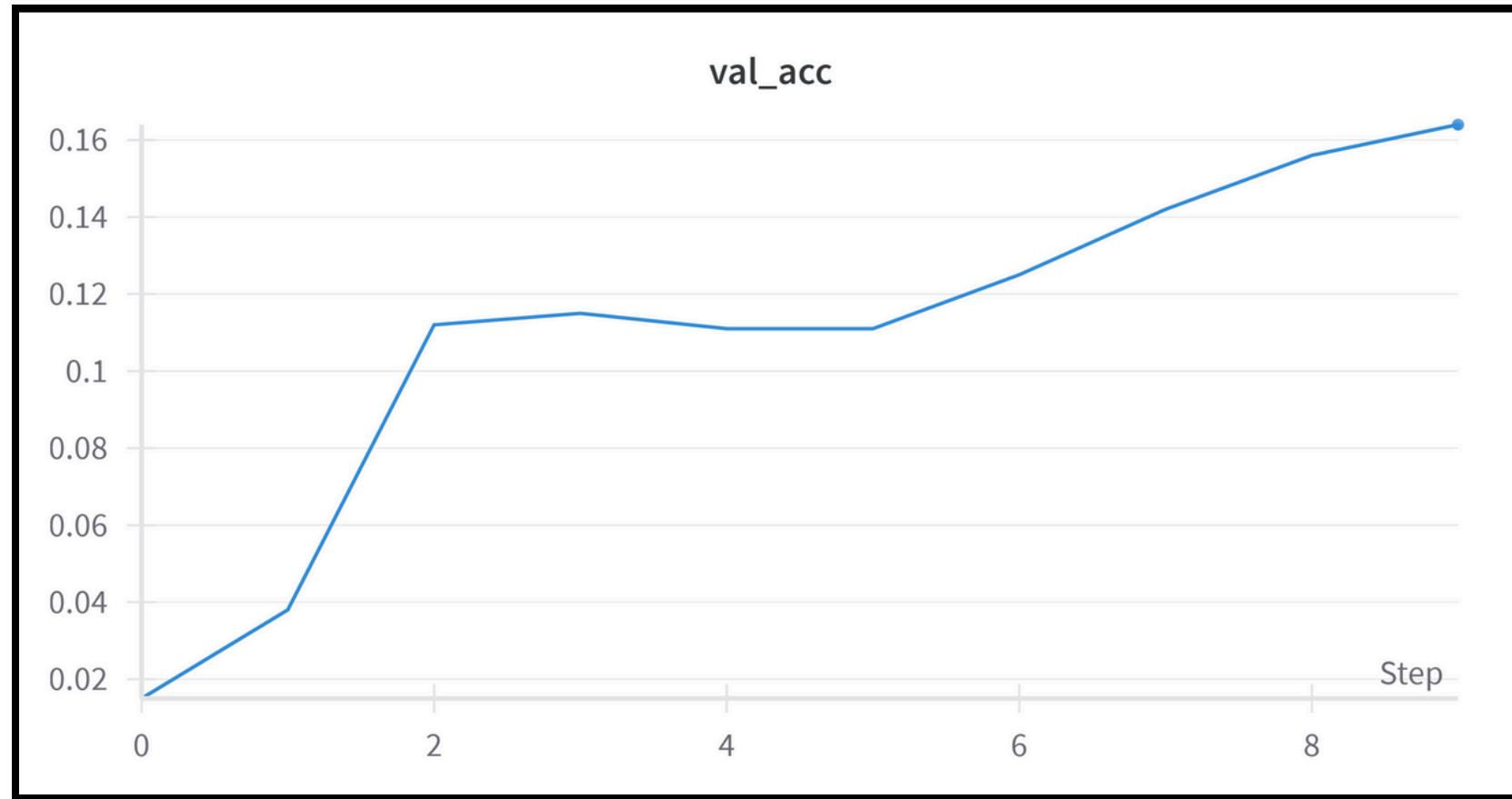


This model when provided with non-contextual RoBERTa embeddings is able to achieve 16.4%.

Baseline: LSTM as Encoder & Decoder

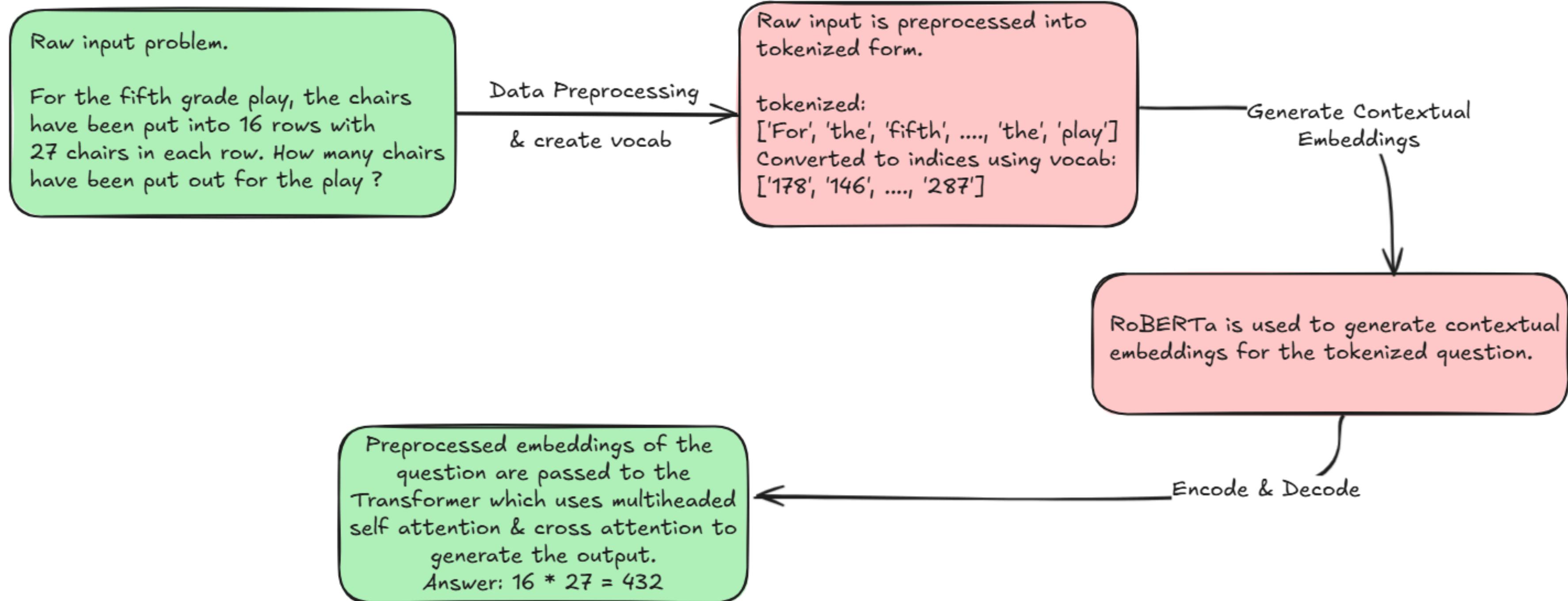


Baseline: LSTM Encoder & Decoder

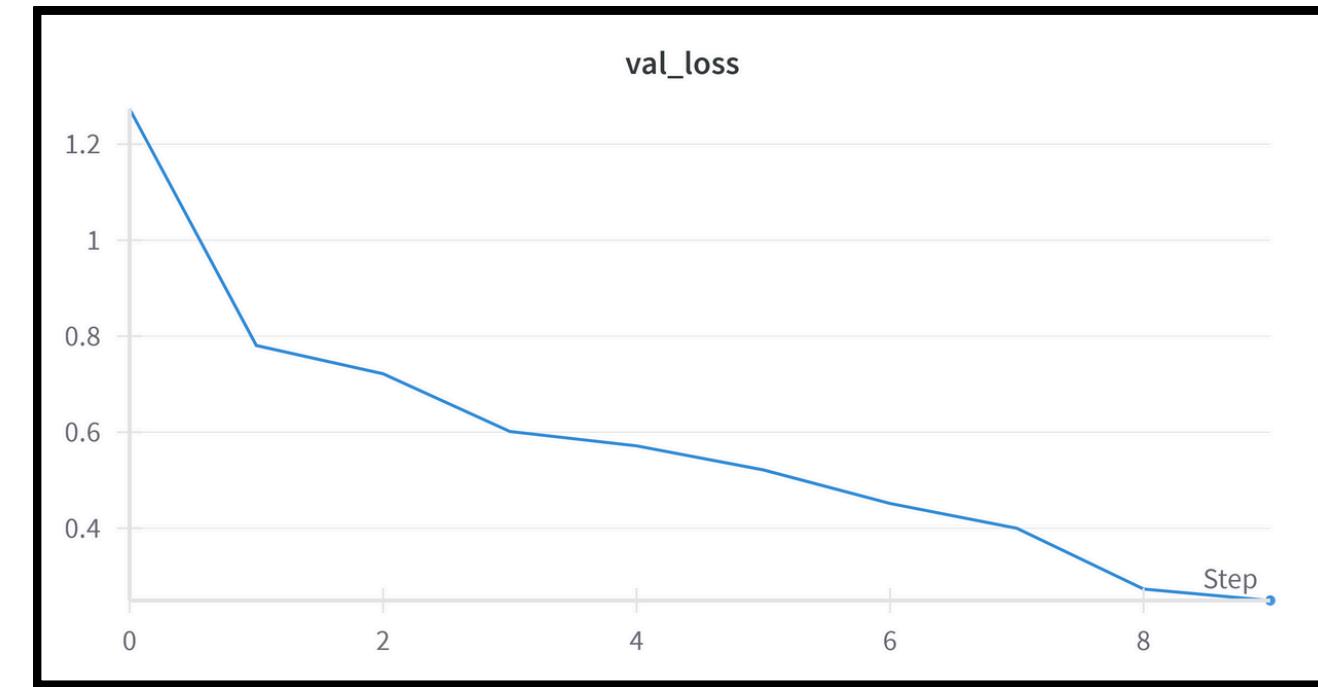
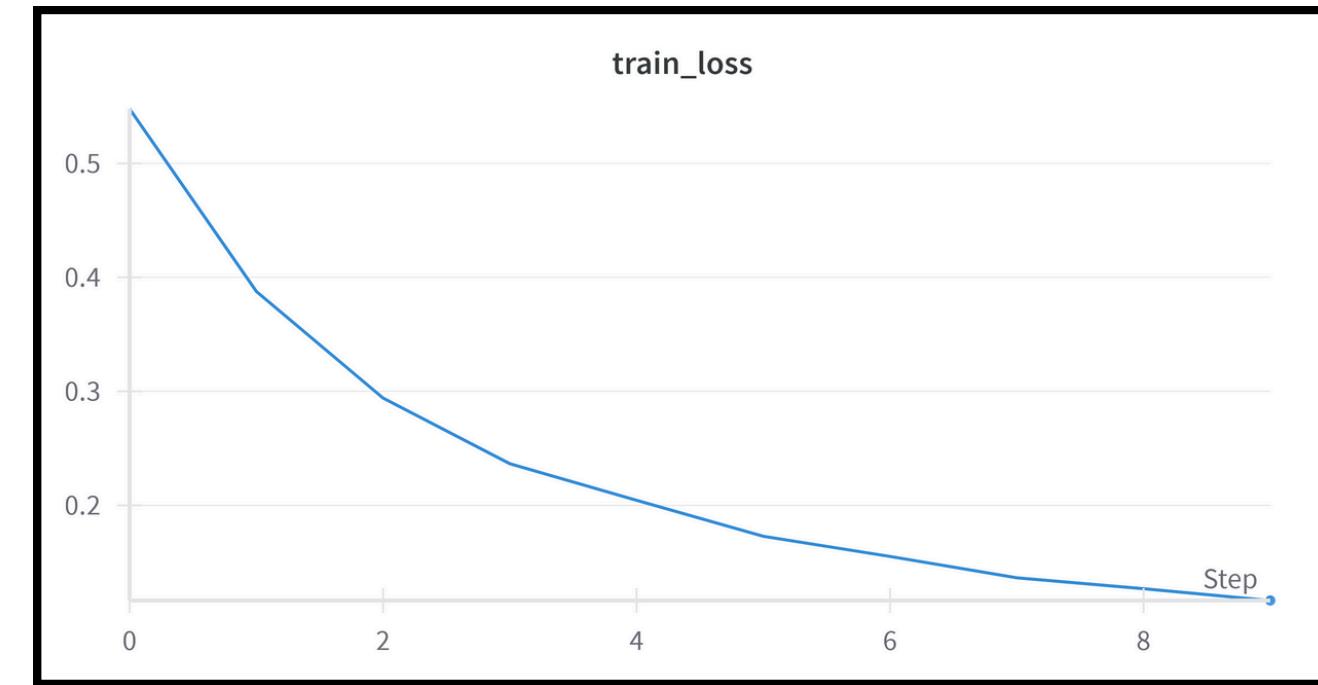
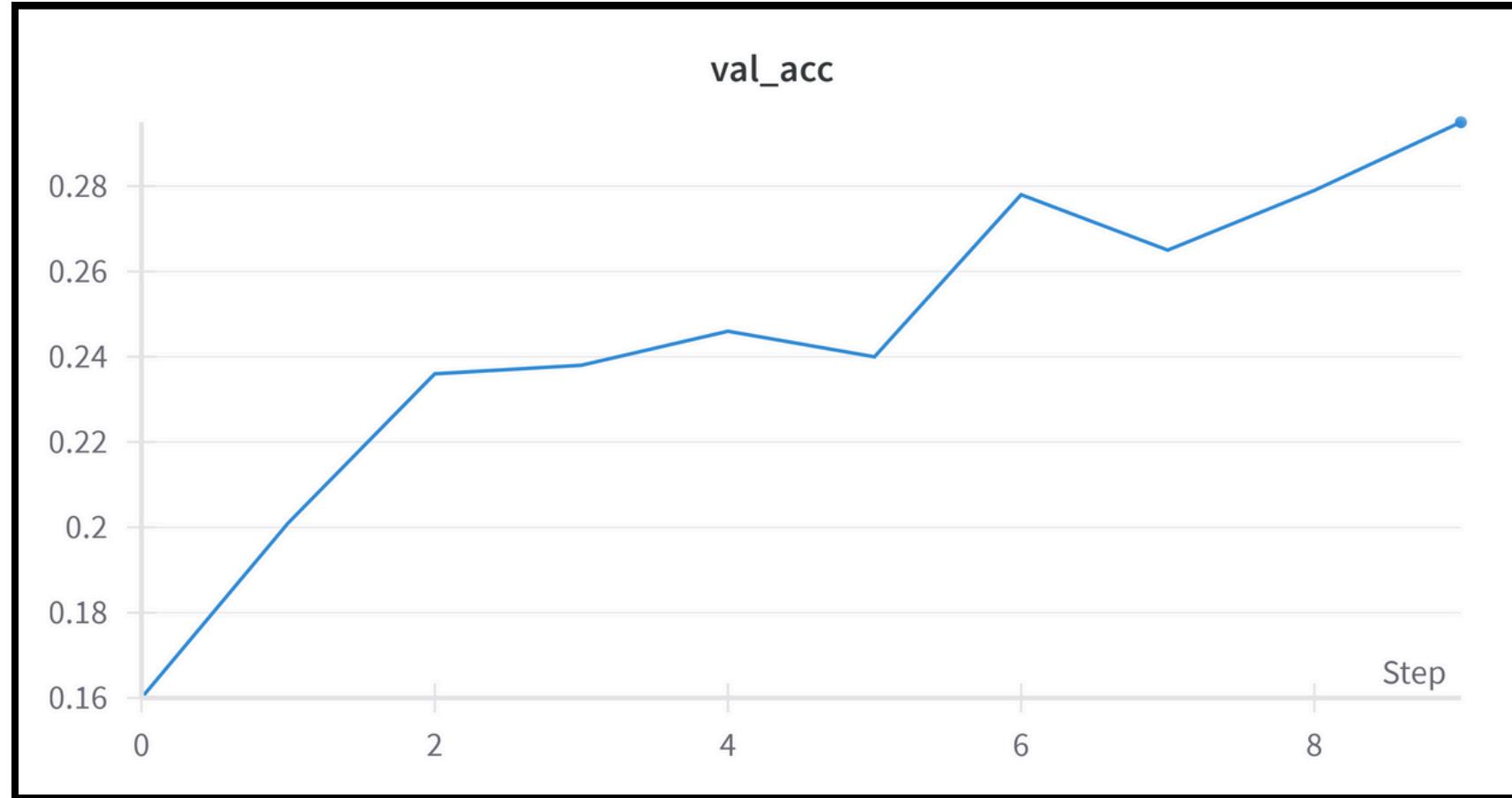


We obtain an accuracy of 20.7% on the validation data of the dataset.

Baseline: Transformer

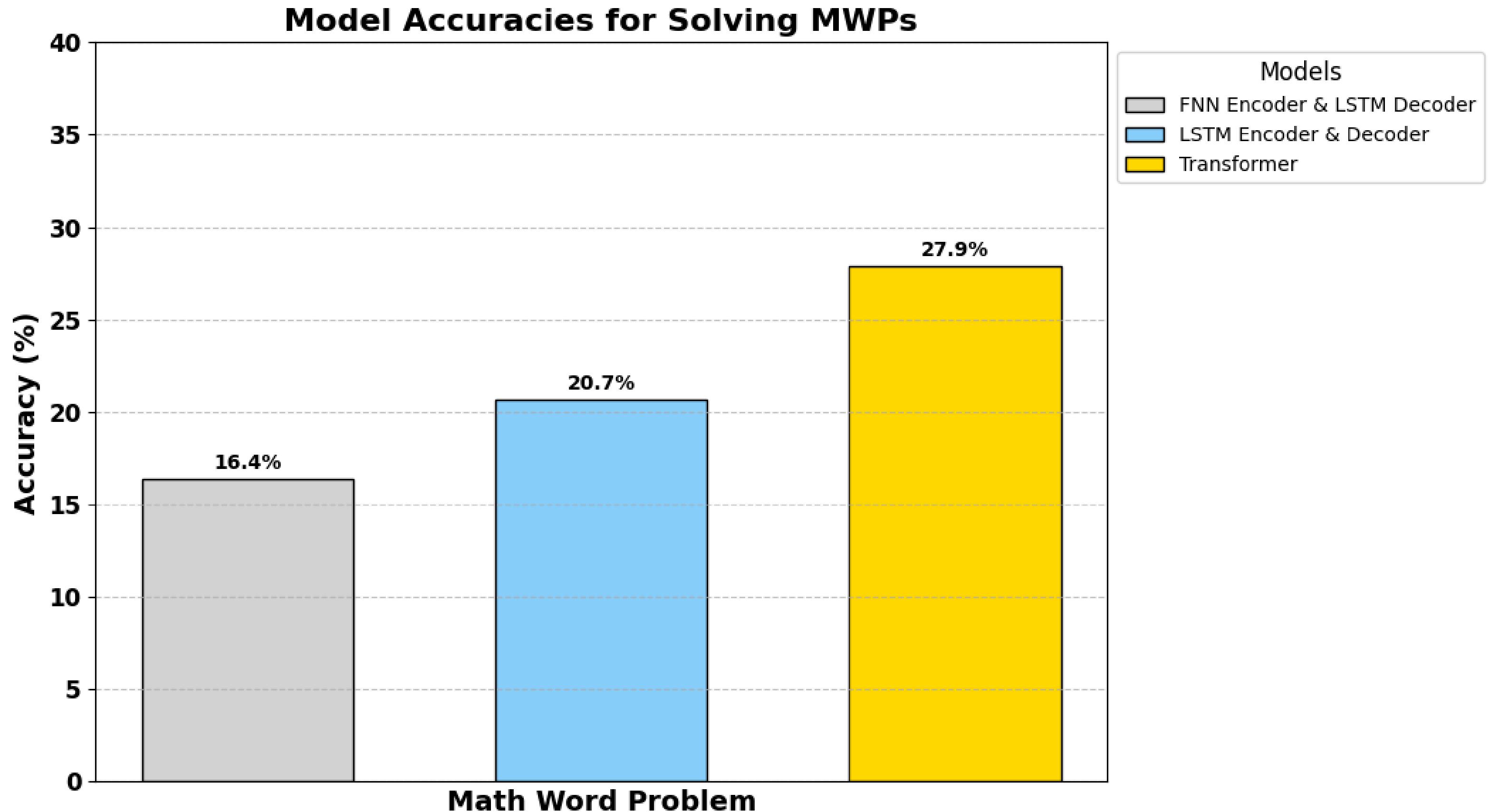


Baseline: Transformer



This model achieves 27.9% accuracy on the ASDiv-A dataset, outperforming LSTM-based method.

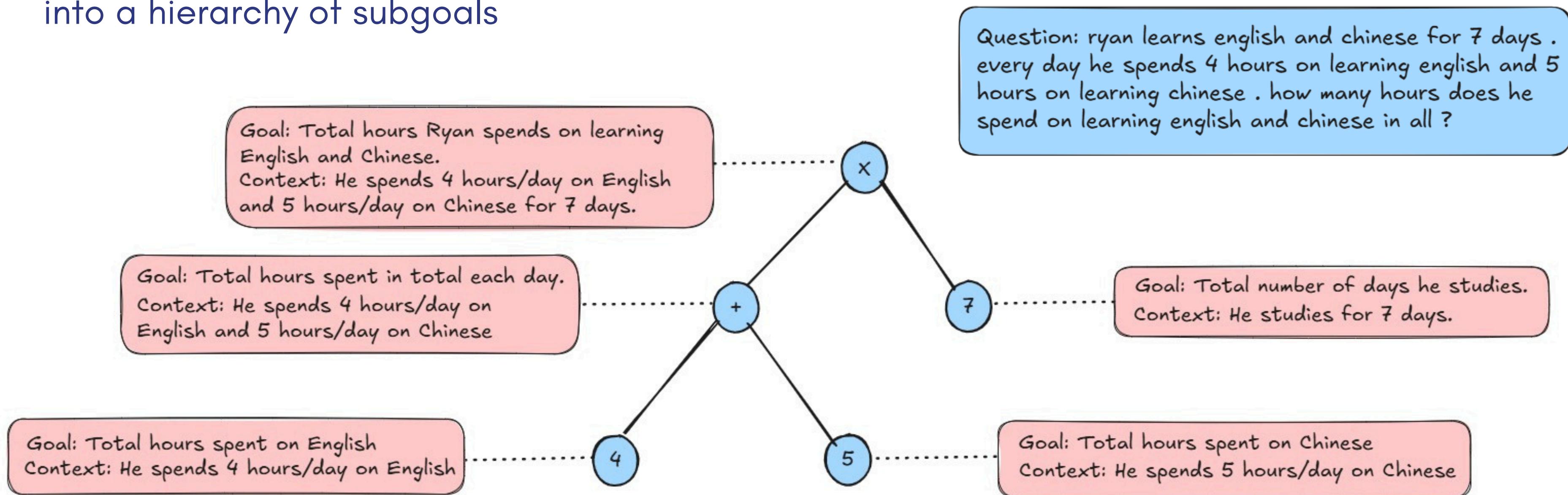
Comparison across baseline models:



Task Specific Models

Goal-Driven Tree-Structured (GTS)

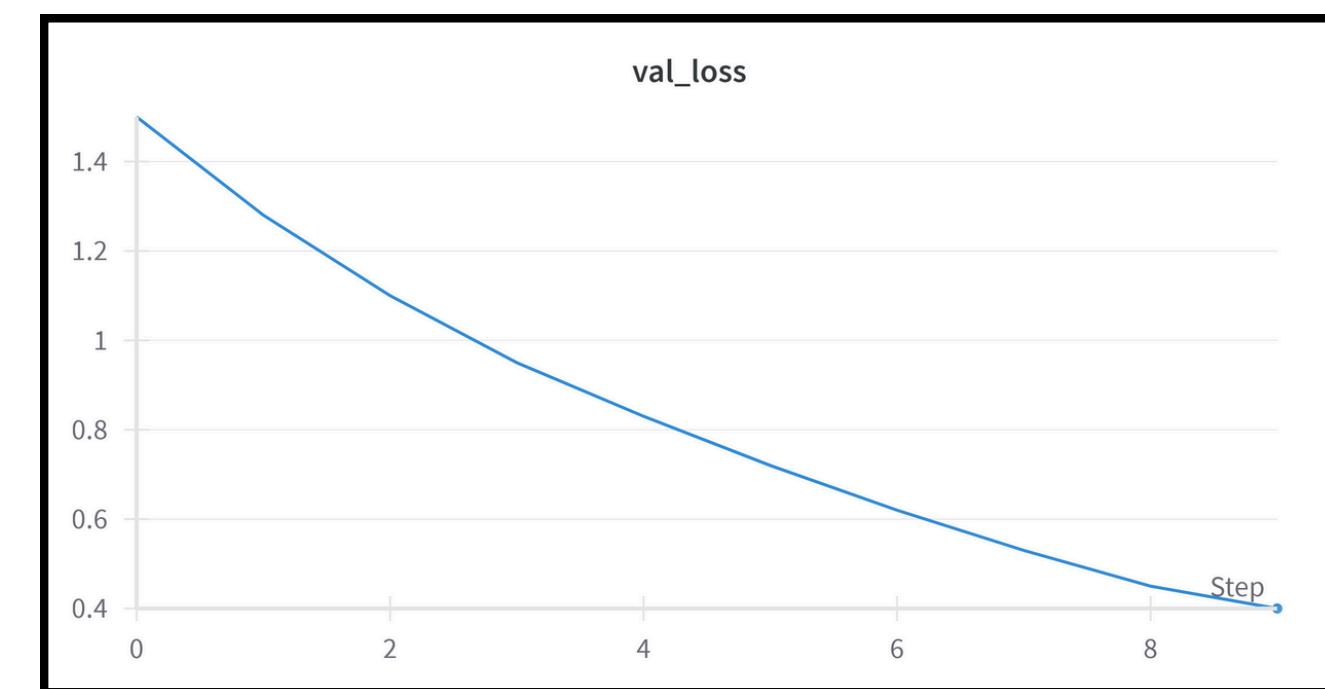
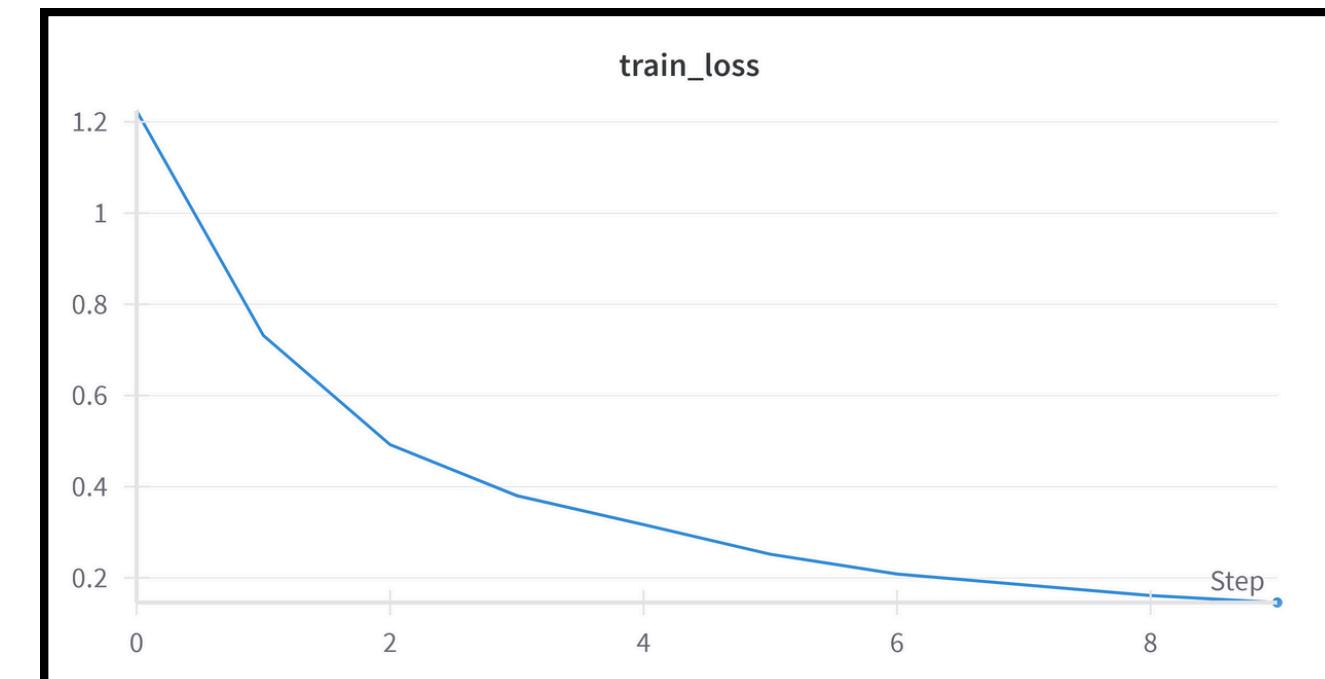
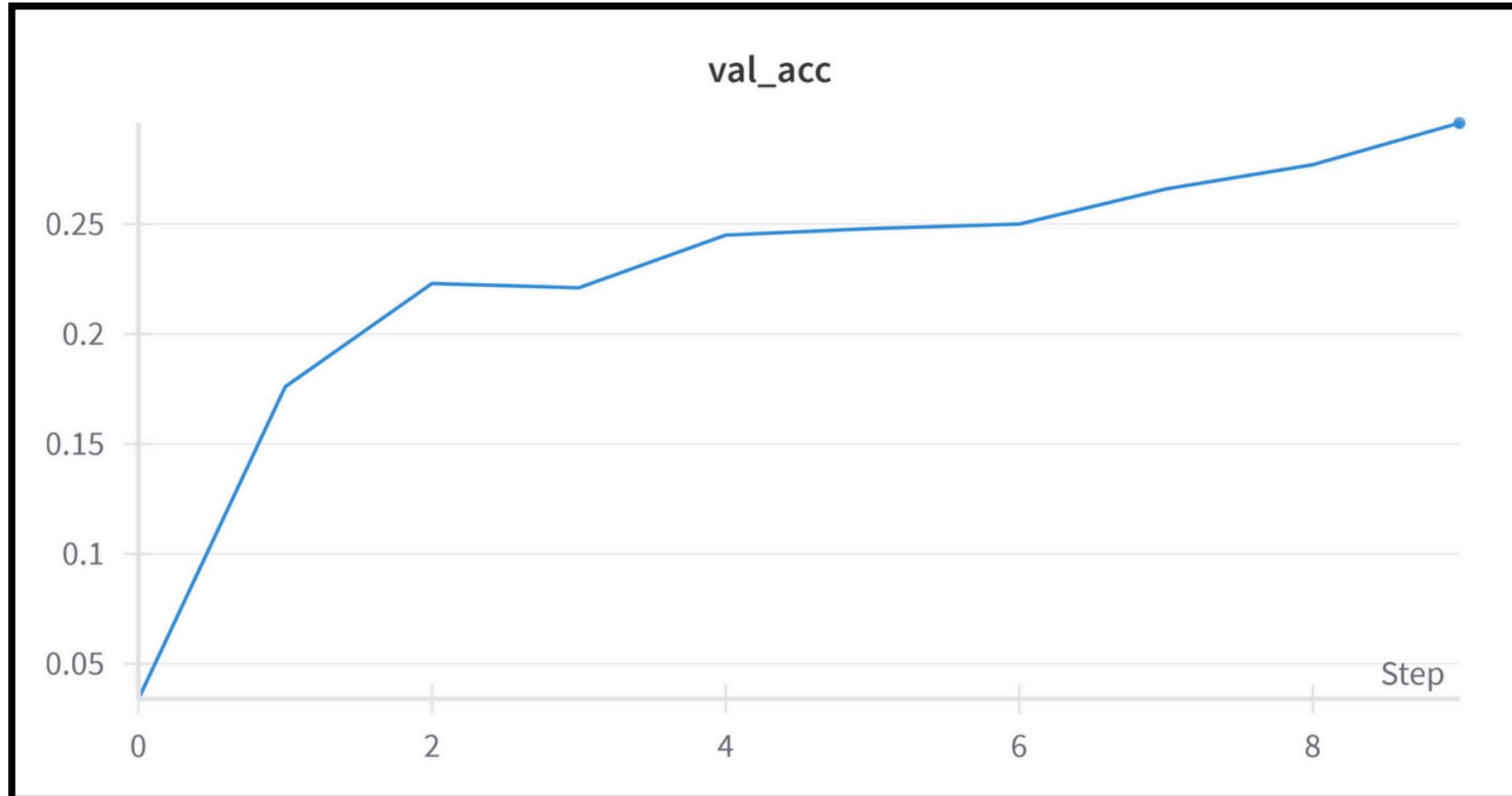
Unlike sequence-to-sequence models like RNNs and Transformers, which generate solution expressions from left to right, the GTS model decomposes the problem into a hierarchy of subgoals



Goal-Driven Tree-Structured (GTS)

- **Encoder:** Processes problem text word by word using a bidirectional GRU.
- **Decoder:** Initializes root goal vector using the final hidden states of the encoder.
- **Top-Down Goal Decomposition:** Uses attention to calculate a context vector summarizing the problem text. Predicts the token with the highest probability.
- **Subgoal Generation:** Combines parent goal, context vector, and predicted token embedding via a two-layer gated feedforward network.
- **Subtree Embedding:** Encodes subtrees bottom-up with a recursive neural network.

Goal-Driven Tree-Structured (GTS)

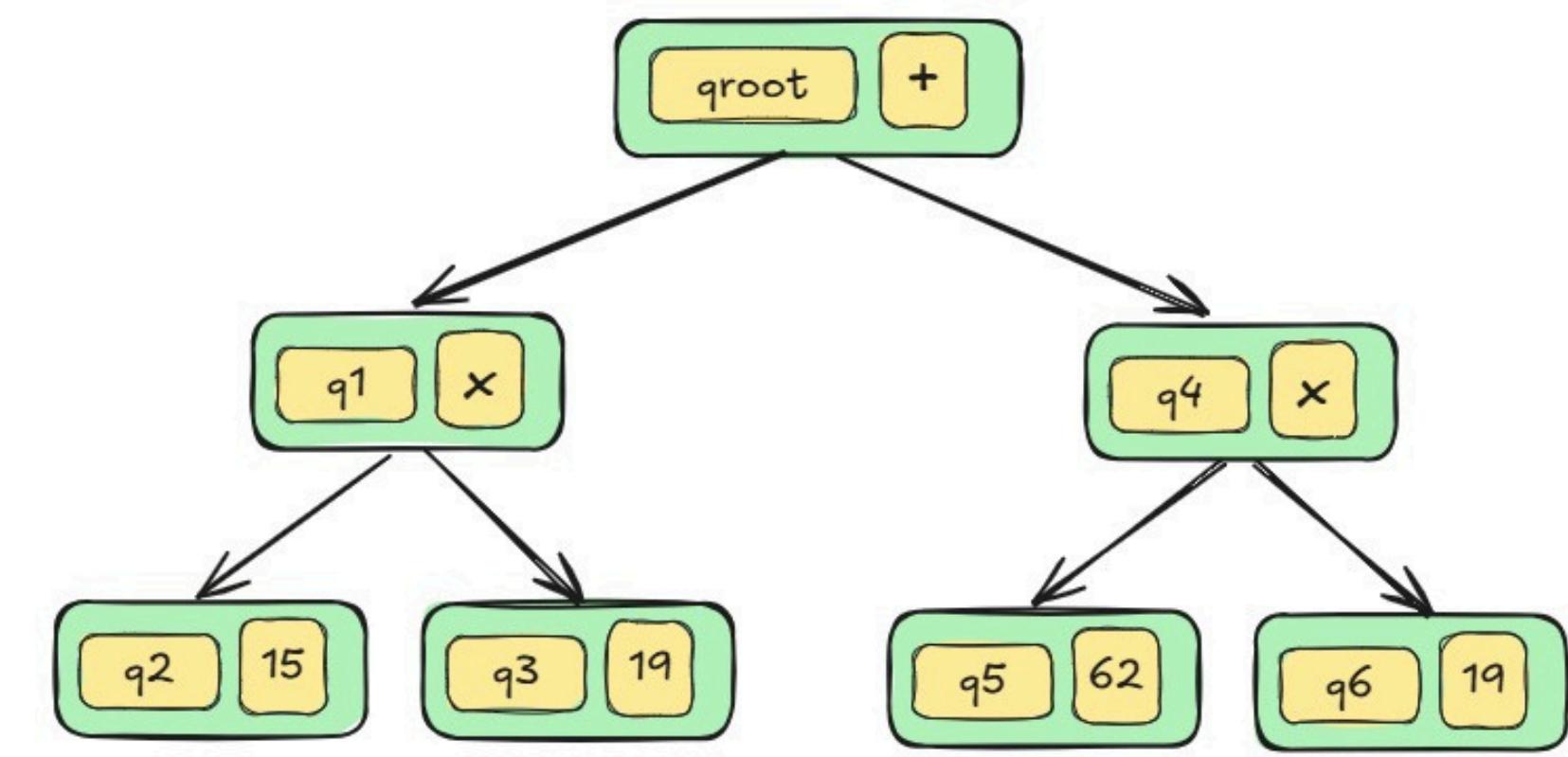
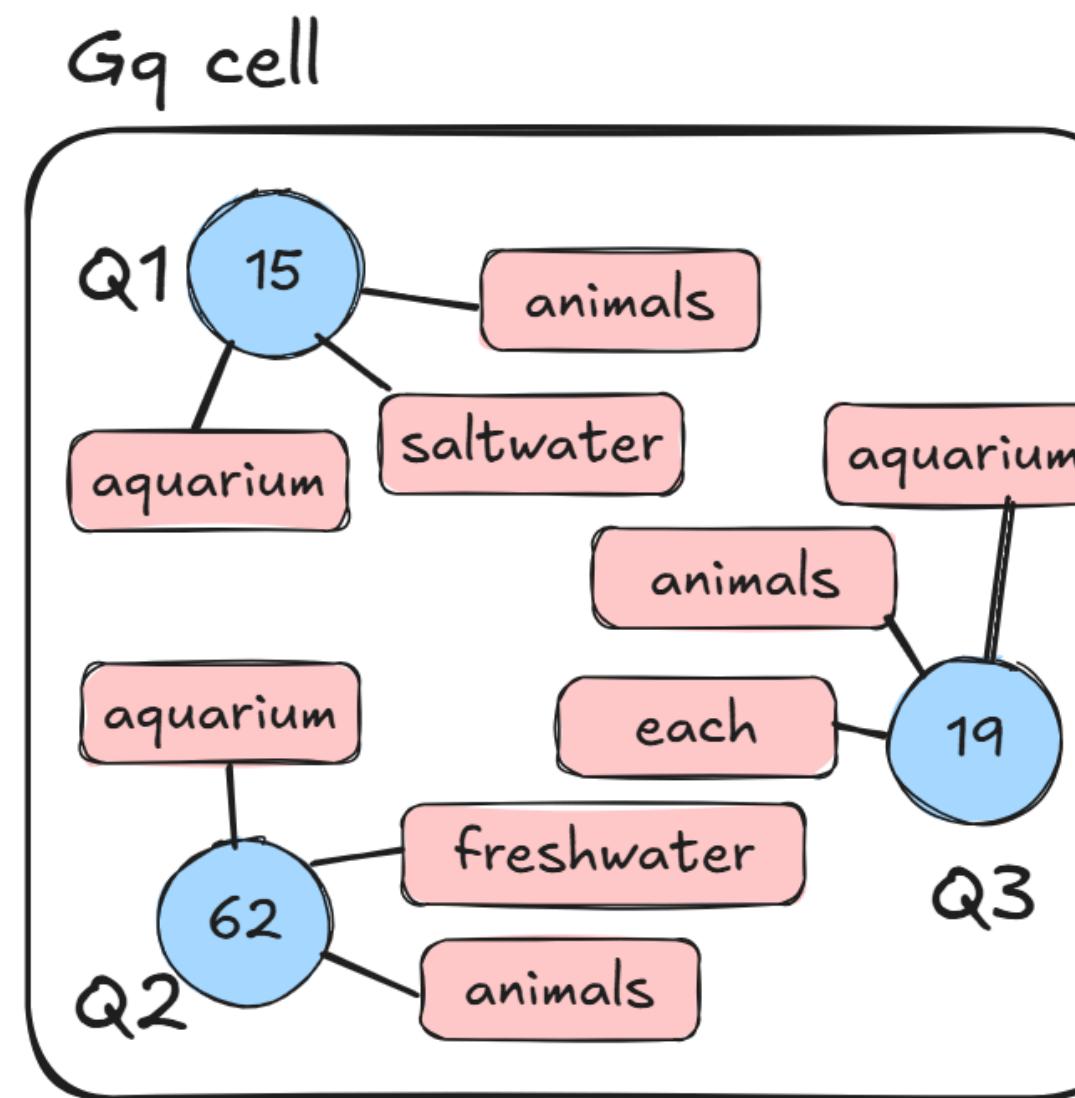
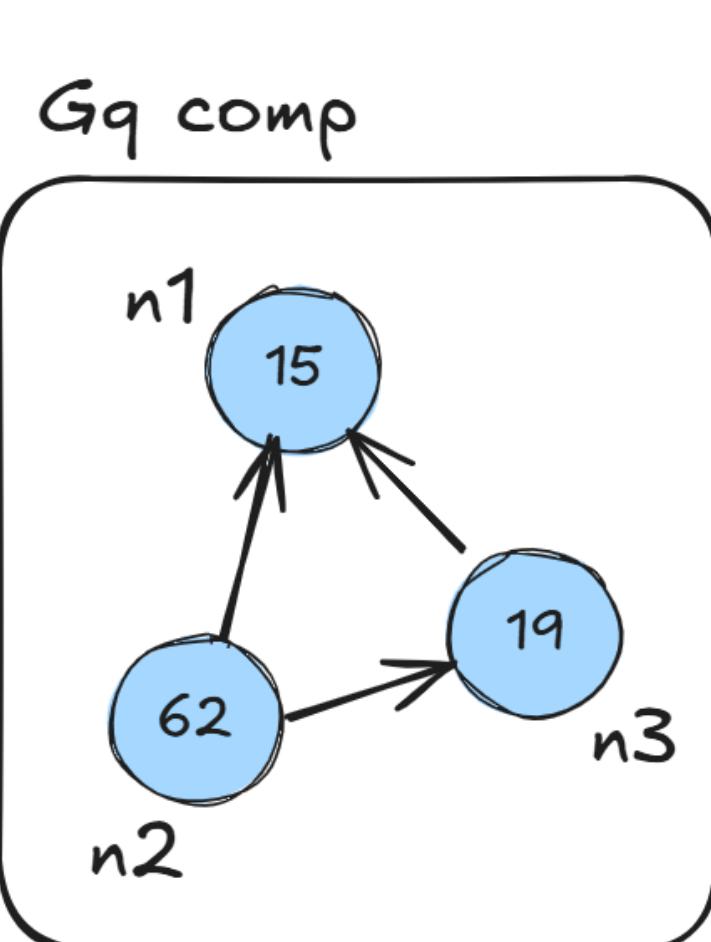


The GTS model solves problems intuitively by breaking them into smaller steps, creating clear and logical tree structures that connect numbers and operations. It achieves an accuracy of 29.6% after 10 epochs

Graph to Tree learning

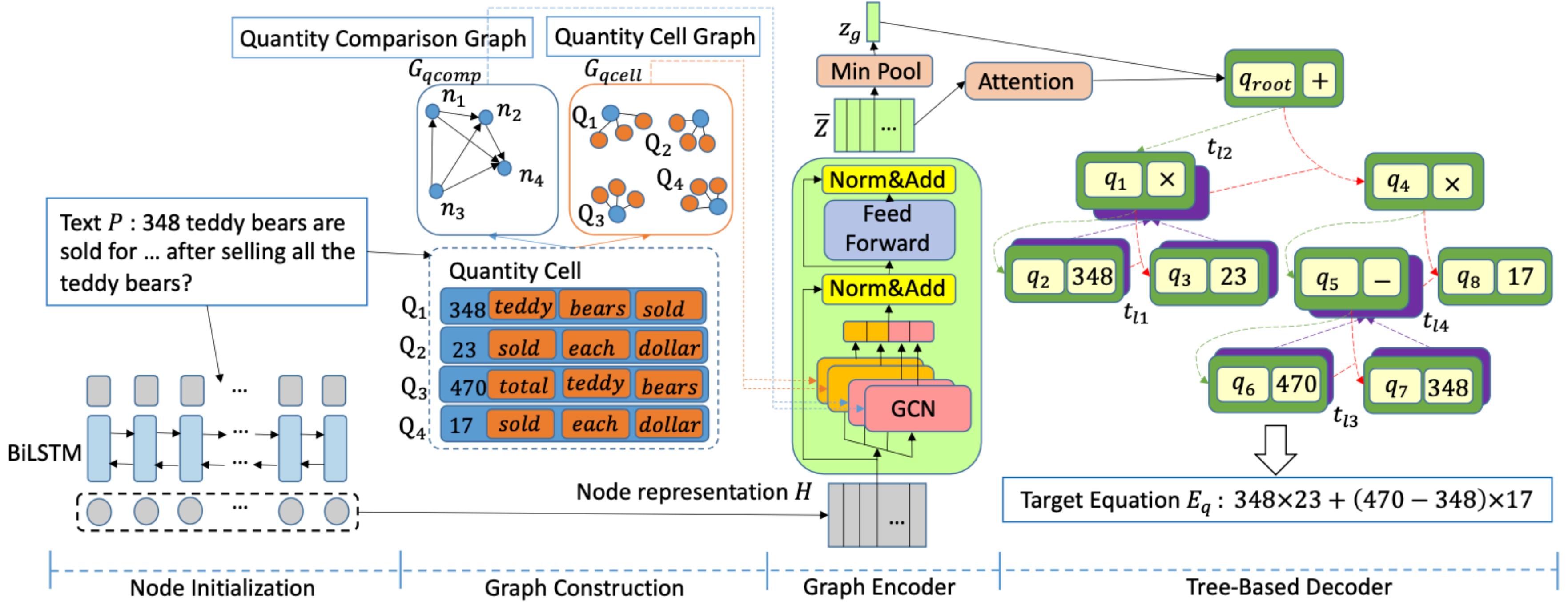
The Graph2Tree model is an extension of the GTS approach that improves the representation of Math Word Problems (MWPs) through graph-based encoding.

He had 15 aquariums for saltwater animals and 62 aquariums for freshwater animals . each aquarium has 19 animals in it . How many animals does he have in total ?



Target Equation Eq : $15 \times 19 + 62 \times 19$

Graph to Tree learning



Graph to Tree learning

Graph2Tree improves GTS performance by enriching problem representation using semantic and numerical relationships between quantities.

Key Graph Structures:

- **Quantity Cell Graph:**

- Connects each quantity to its corresponding nouns, adjectives, verbs, units, and rates.
- Extracted through dependency parsing, constituency parsing, and POS tagging.
- Example: Links "15" to "aquariums" within the context of "saltwater animals."

- **Quantity Comparison Graph:**

- Represents numerical relationships through directed edges based on magnitude.
- Prevents generating unrealistic expressions (e.g., subtracting larger numbers from smaller ones).

Graph to Tree learning

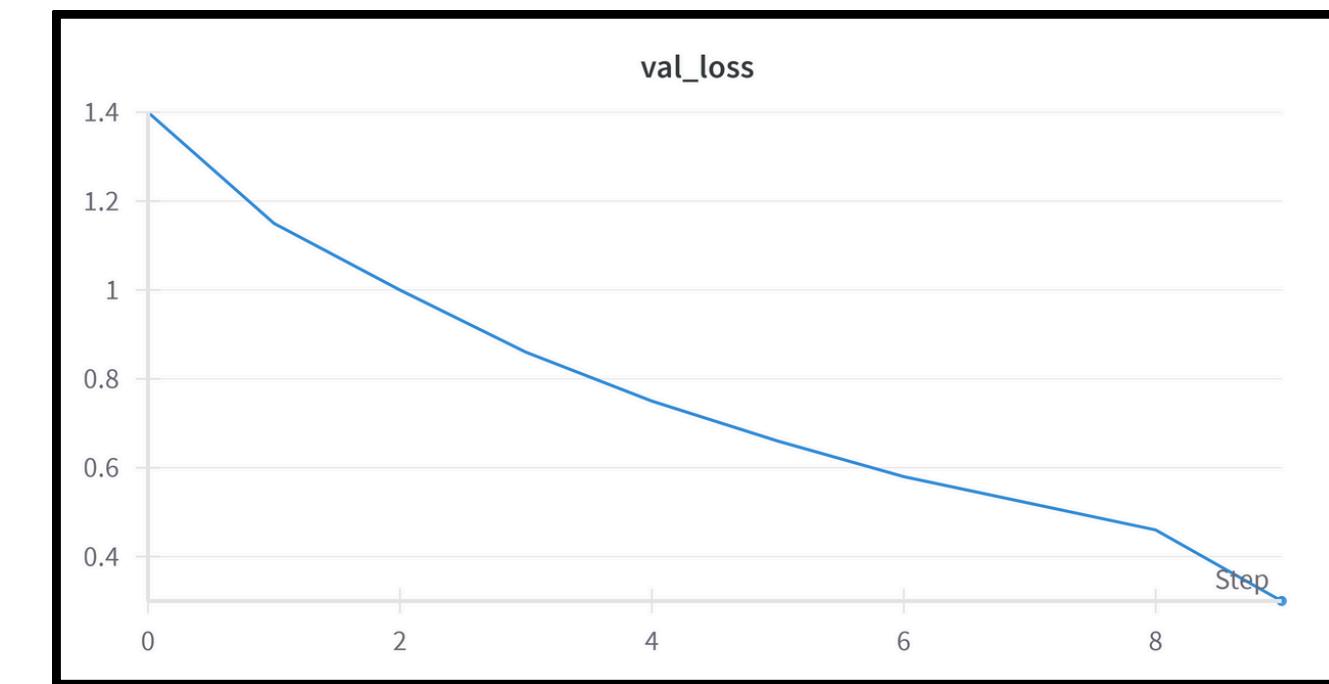
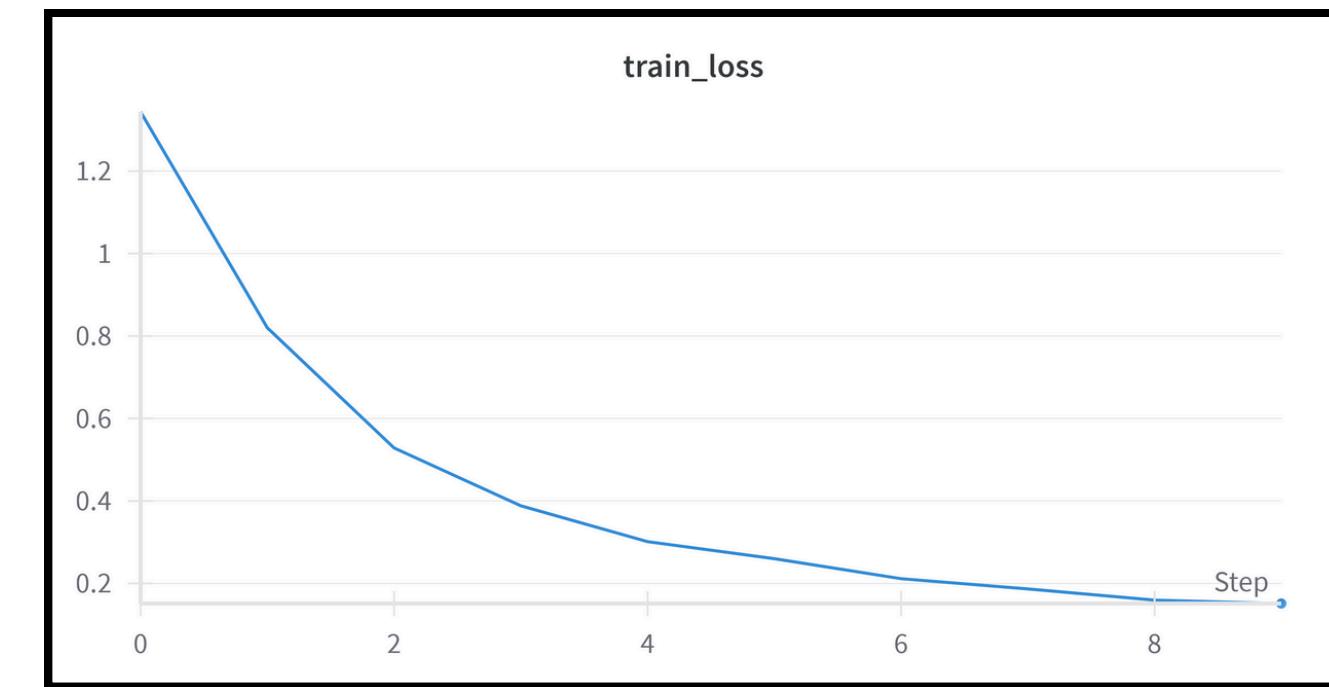
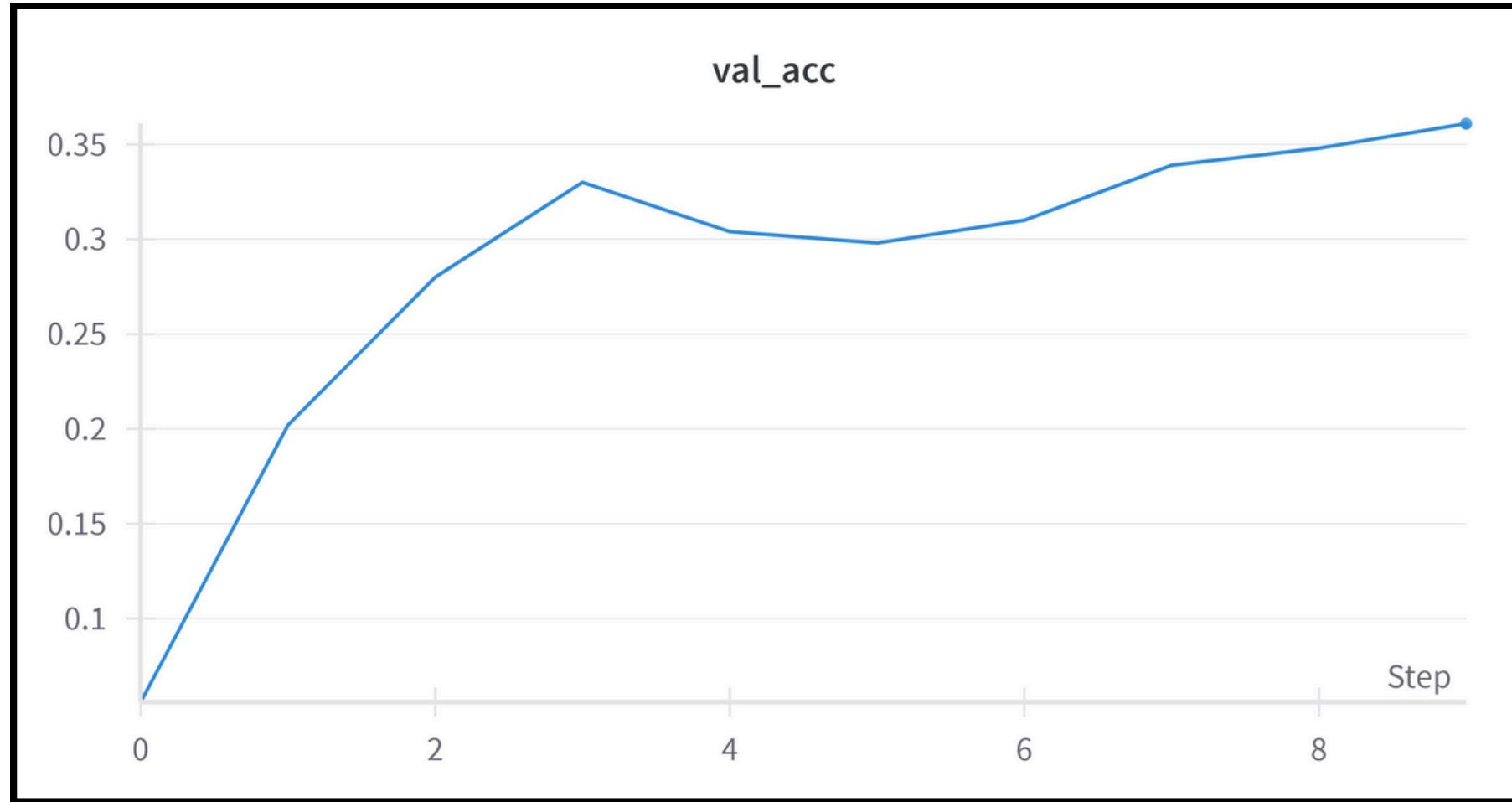
- **Graph Transformer**

- Employs GCNs, adjacency matrices, and initial node embeddings to capture graph relationships.
- Produces enhanced node representations along with a global graph representation that embodies the semantic and numerical connections of the issue.

- **Tree-Based Decoder**

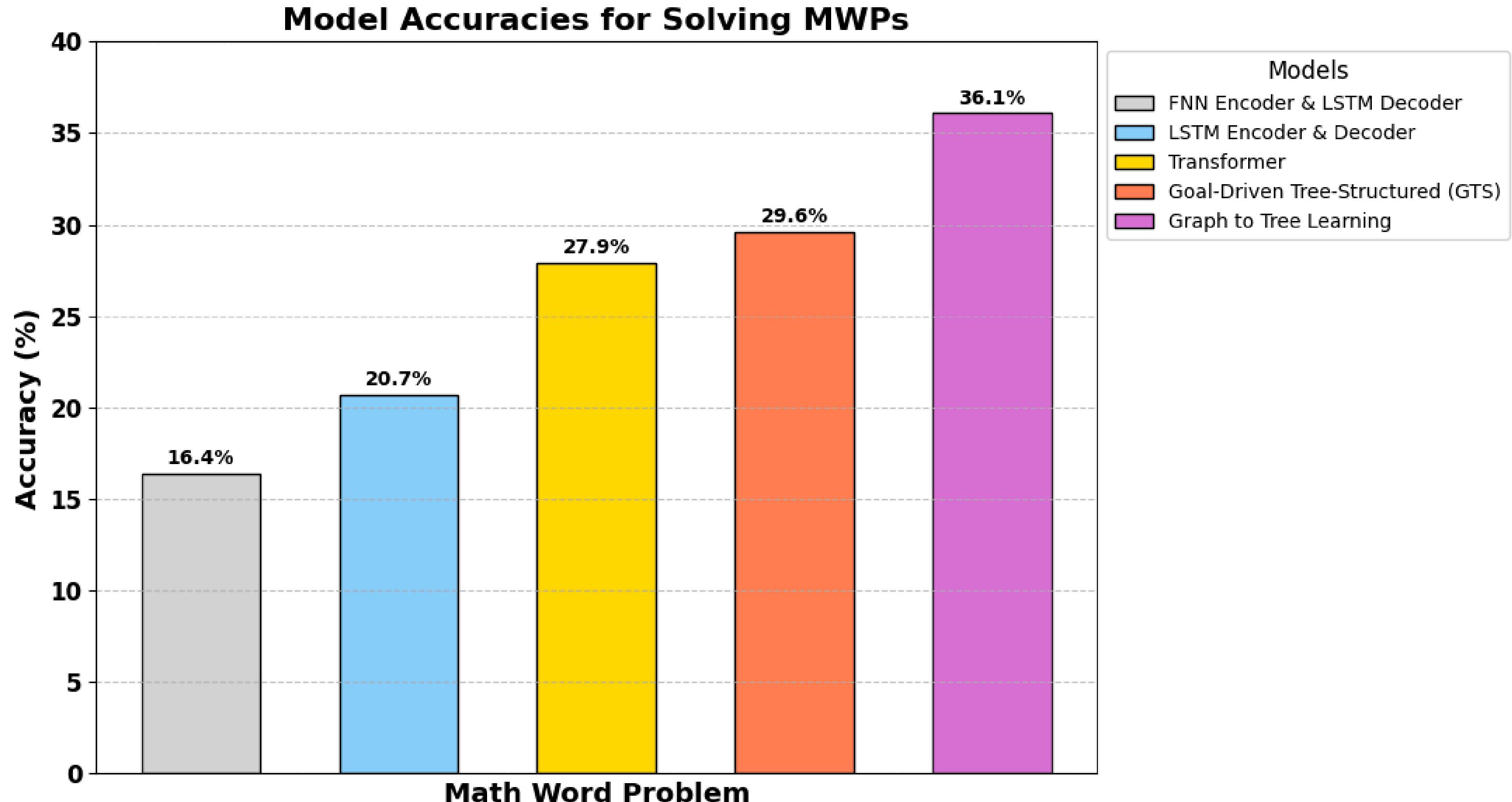
- This method creates solution expressions depicted as trees using pre-order traversal using GTS Tree based-decoder.
- It begins with the root node defined by the global graph representation, then recursively develops left and right child nodes by integrating subtree embeddings.

Graph to Tree learning



By enriching the representation of the problem with these two graphs, Graph2Tree provides the GTS decoder with more informative input, allowing it to generate more accurate and realistic solution expressions, achieving an accuracy of 36.1% after 10 epochs.

Analysis



LLMs

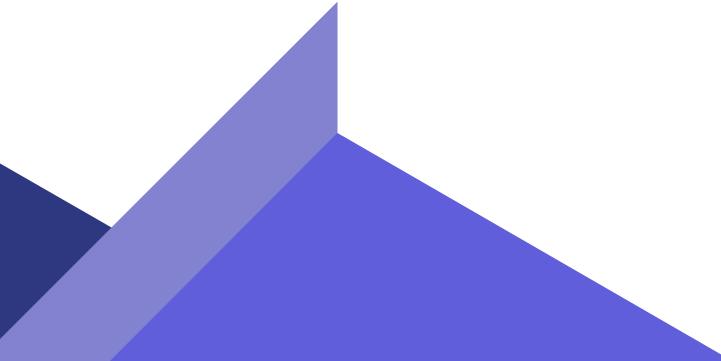
LLMs

Math word problems (MWPs) have been proposed as a challenging testbed for LLMs, as they test not only the ability of the models to deal with purely mathematical expressions, but also their reasoning and natural language understanding abilities.

By investigating LLM performance on MWPs, we aim to gain deeper insights into their reasoning capabilities and identify ways to enhance their problem-solving skills.

We want to do an analysis of the performance of LLMs and try to understand how well the model performs with different prompting techniques like: chain-of-thought prompting, few-shot prompting, zero-shot prompting, and how the performance scales as the complexity of problems increases.

Gemini-2-9B



Gemini-2-9B

- The direct evaluation of the pre-trained Gemma-2-9B (2024) was conducted on the MAWPS-ASDIV dataset without any fine-tuning. This evaluation aimed to understand the model's baseline performance in solving mathematical word problems.
- Prompting Strategy: To guide the model's response format, the following prompt was used for each problem:

Solve the following math problem. Provide only the numerical answer in the format: "the answer is [number]" without any explanation.

- The above ensured consistent output formatting, for evaluating model accuracy. The generated responses were compared against the ground truth answers for accuracy evaluation. This model achieved an accuracy of approximately 64% on the dataset.

Gemini-2-9B

Fine-Tuning using Standard Prompting

- To improve the baseline performance of Gemma-2-9B on the MAWPS-ASDIV dataset, we employed fine-tuning with domain-specific training examples.
- We utilized Low-Rank Adaptation (LoRA), which inserts trainable low-rank matrices into certain layers of the model for fine-tuning. The model is loaded with 4-bit quantization, reducing memory usage. The same prompting strategy as before was used.
- After fine-tuning, the model achieved an accuracy of 67% on the MAWPS-ASDIV data.

Mistral Instruct 7B

- The direct evaluation of the pre-trained Mistral Instruct 7B (2024) was conducted on the MAWPS-ASDIV dataset without any fine-tuning. This evaluation aimed to understand the model's baseline performance in solving mathematical word problems.
- Prompting Strategy: Same as Gemma-2-9B.
- The model accuracy before any finetuning was 40%.
- The model was then finetuned in a similar regime as done for Gemma-2-9B via standard prompting. The model's performance was then noted to be 56% on the test data.



Gemini-2-9B

Fine-Tuning using Few Shot Prompting

- We presented a few task-specific examples within the training prompts to guide the model's understanding of the problem format and expected responses. Format:

Here are a few examples for your reference:

- Question: Tom has \$100. He spends 40% of the money on books. How much money does he have left now?
Answer: the answer is 60.0

- The fine-tuned model was evaluated on 50 samples from the MAWPS-ASDIV dataset due to lack of efficient compute. The results indicated an accuracy of 88%, reflecting a improvement over previous methods.
- This strategy not only provided contextual guidance to the model but also improved its understanding of task-specific nuances.

Standard Prompting

Model Input

Q. Julia played tag with 18 kids on Monday . She played tag with 10 kids on Tuesday. How many more kids did she play with on Monday than on Tuesday ?

A. The answer is 8. ✓

Q. Bobby ate 17 pieces of candy . Then he ate 15 more . If he initially had 36 pieces of candy how many pieces of candy does he still have left ?

Model Output

A. The answer is 8. ✗

Few-Shot Prompting

Model Input

Q. Julia played tag with 18 kids on Monday . She played tag with 10 kids on Tuesday. How many more kids did she play with on Monday than on Tuesday ?

Here are some examples for your reference:

A recipe calls for 2.5 cups of flour to make 1 dozen cookies. How many cups of flour are needed to make 3 dozen cookies?

The answer is 7.5.

...

A. The answer is 8. ✓

Q. Bobby ate 17 pieces of candy . Then he ate 15 more . If he initially had 36 pieces of candy how many pieces of candy does he still have left ?

Here are some examples for your reference:

...

Model Output

A. The answer is 4. ✓

Gemini-2-9B

Inference-Side Chain-Of-Thought (CoT) Prompting

- Chain-of-Thought (CoT) prompting was applied to the Gemma-2-9B model during inference to improve its reasoning capabilities when solving mathematical word problems.
- CoT prompting encourages the model to generate step-by-step reasoning before arriving at the final answer, using intermediate reasoning paths to enhance consistency.
- To implement CoT prompting, the model was provided with the instruction to "think step-by-step" before producing the final numerical answer.
- Accuracy of 94% over the same 50 samples was observed.

Few-Shot Prompting

Model Input

Q. Julia played tag with 18 kids on Monday . She played tag with 10 kids on Tuesday. How many more kids did she play with on Monday than on Tuesday ?

Here are some examples for your reference:
A recipe calls for 2.5 cups of flour to make 1 dozen cookies. How many cups of flour are needed to make 3 dozen cookies?

The answer is 7.5.

...

A. The answer is 8. ✓

Q. Bobby ate 38 pieces of candy . then he ate 36 more . He also ate 16 pieces of chocolate . How many more pieces of candy than chocolate did Bobby eat ?

Here are some examples for your reference:

...

Model Output

A. The answer is 2. ✗

CoT Prompting

Model Input

Q. Julia played tag with 18 kids on Monday . She played tag with 10 kids on Tuesday. How many more kids did she play with on Monday than on Tuesday ?

A. Let's think step by step:

1. We know that Julia played tag with 18 kids on Monday and 10 kids on Tuesday.
2. To find out how many more kids she played with on Monday than on Tuesday, we need to subtract the number of kids she played with on Tuesday from the number of kids she played with on Monday.
3. So, the calculation is: $18 - 10 = 8$.
4. Therefore, the answer is 8.

The answer is 8. ✓

Q. Bobby ate 38 pieces of candy . then he ate 36 more . He also ate 16 pieces of chocolate . How many more pieces of candy than chocolate did Bobby eat ?

Reasoning

Model Output

Model Output

A. Let's think step by step:

1. Bobby ate 38 pieces of candy.
2. He ate 36 more pieces of candy.
3. He also ate 16 pieces of chocolate.
4. To find out how many more pieces of candy he ate than chocolate, we need to subtract the number of chocolate pieces from the total number of candy pieces.
5. Total number of candy pieces = $38 + 36 = 74$
6. Number of chocolate pieces = 16
7. Number of more candy pieces = $74 - 16 = 58$

The answer is 58. ✓

Reasoning

Gemini-2-9B

Chain-Of-Thought (CoT) + Few-Shot Prompting

- Chain-of-Thought (CoT) reasoning combined with few-shot prompting was used during inference to evaluate the Gemma-2-9B model's performance on the dataset.
- The CoT + Few-Shot prompting method utilized a multi-part prompt, beginning with explicit instructions to "think step-by-step" for reasoning through each problem. The prompt also included several few-shot examples of solved problems, formatted to demonstrate the concise step-by-step reasoning process alongside the answer.
- This combined strategy was evaluated on the MAWPS-ASDIV dataset, achieving an accuracy of 96%. This shows an improvement over both standalone CoT and few-shot prompting techniques, highlighting the positive effect of combining these approaches.

CoT Prompting

Model Input

Q. Julia played tag with 18 kids on Monday . She played tag with 10 kids on Tuesday. How many more kids did she play with on Monday than on Tuesday ?

A. Let's think step by step:

1. We know that Julia played tag with 18 kids on Monday and 10 kids on Tuesday.
2. To find out how many more kids she played with on Monday than on Tuesday, we need to subtract the number of kids she played with on Tuesday from the number of kids she played with on Monday.
3. So, the calculation is: $18 - 10 = 8$.
4. Therefore, the answer is 8.

The answer is 8. ✓

Q. Allan brought 7 balloons and 5 balls while Jake brought 6 balloons and 4 balls to the park . how many balls did Allan and Jake have in the park ?

Model Output

A. Let's think step by step:

- Allan brought 7 balloons and 5 balls, so he has a total of $7 + 5 = 12$ items.
Jake brought 6 balloons and 4 balls, so he has a total of $6 + 4 = 10$ items.
Therefore, the answer is $12 + 10 = 22$.

The answer is 22. ✗

CoT + Few-Shot Prompting

Model Input

Q. Julia played tag with 18 kids on Monday . She played tag with 10 kids on Tuesday. How many more kids did she play with on Monday than on Tuesday ?

Here are some examples:

Example-1:

Input:

A store offers a 20% discount on a \$80 shirt. what is the final price after discount ?

Response:

Let's think step by step:

1. Calculate the discount amount: $\$80 \times 20\% = \$80 \times 0.2 = \$16$.
2. Subtract discount from original price: $\$80 - \$16 = \$64$.

Therefore, the answer is 64.

A. Let's think step by step:

1. Kids played with on Monday = 18
2. Kids played with on Tuesday = 10
3. Kids played more on Monday than on Tuesday = Kids on Monday - Kids on Tuesday = $18 - 10$

Therefore, the answer is 8. ✓

Q. Allan brought 7 balloons and 5 balls while Jake brought 6 balloons and 4 balls to the park . how many balls did Allan and Jake have in the park ?

Here are some examples:

...

Model Output

A. Let's think step by step:

1. Number of balls Allan brought = 5
 2. Number of balls Jake brought = 4
 3. Total number of balls in the park = Number of balls Allan brought + Number of balls Jake brought = $5 + 4$
- Therefore, the answer is 9

The answer is 9. ✓

Few shot examples with reasoning

Conclusion

CoT + Few-Shot Prompting

Model Input

Q. Bobby had 22 pieces of candy . He ate some pieces of candy . Then he ate 5 more . If he still has 8 pieces of candy left how many pieces of candy had he eaten at the start ?

Here are some examples:

Example-1:

Input:

A store offers a 20% discount on a \$80 shirt. what is the final price after discount ?

Response:

Let's think step by step:

1. Calculate the discount amount: $\$80 \times 20\% = \$80 \times 0.2 = \$16$.
2. Subtract discount from original price: $\$80 - \$16 = \$64$.

Therefore, the answer is 64.

Model Output

A. Let's think step by step:

1. Bobby had 22 pieces of candy.
 2. He ate some pieces of candy.
 3. He ate 5 more pieces of candy.
 4. He still has 8 pieces of candy left.
 5. To find out how many pieces of candy he had eaten at the start, we need to subtract the number of pieces of candy he has left from the total number of pieces of candy he had.
 6. Total number of pieces of candy he had = 22
 7. Number of pieces of candy he has left = 8
 8. Number of pieces of candy he had eaten at the start = Total number of pieces of candy he had - Number of pieces of candy he has left = $22 - 8 = 14$
- Therefore, the answer is 14. ❌

From our evaluation, we can conclude that LLMs show potential in solving Mathematical Word Problems (MWP)

LLMs still slightly struggle with reasoning for complex problems, even when employing advanced techniques like CoT + Few-shot prompting.

As seen from some examples, it indicates gaps in their ability to handle multi-step complex reasoning or implicit mathematical deductions.

Future Directions

- Combining CoT with External knowledge retrieval to guide reasoning steps.
- Interpretability: Focusing on analyzing where and how the models fail to reason correctly.

Mathematical Word Problem Solving

Swayam Agrawal
CSE
IIIT Hyderabad

Pratham Thakkar
CSE
IIIT Hyderabad

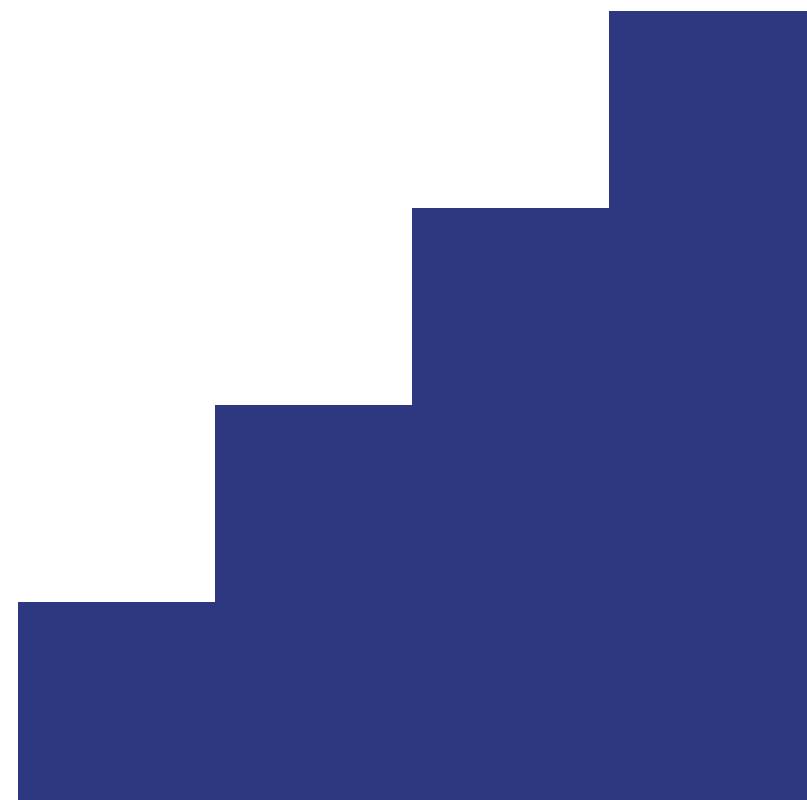
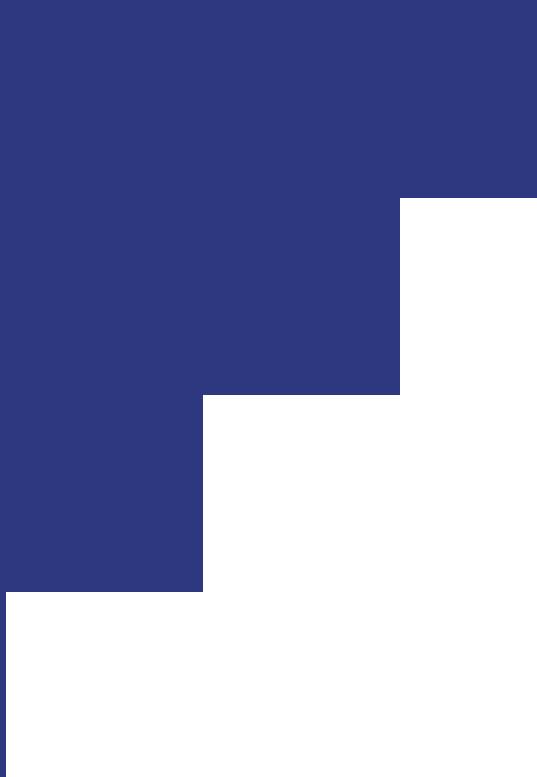
Yash Kawade
CSE
IIIT Hyderabad

Abstract

Mathematical problems are often stated in words in different scenarios, thus requiring problem solvers to extract information from the text and formulate in mathematical language to get the problem's answer. This project explores advanced Natural Language Processing (NLP) approaches to solve MWPs, comparing traditional models like RNNs and Transformers, math-specific architectures like Graph-to-Tree & GTS, and large language models (LLMs) such as Gemma and Mistral. We also evaluate the impact of reasoning paradigms like Chain-of-Thought (COT) prompting.

Our findings highlight trade-offs among these methods: while Graph-To-Tree excels in structured reasoning, LLMs demonstrate strong generalization, especially when fine-tuned or enhanced with Few-shot prompting or COT strategies.

This study underscores the potential of classical NLP approaches and LLMs for MWPs, emphasizing the need for fine-tuning and more complex reasoning strategies.



Demo Video

localhost:8501

old | Brainstel.. Questions - QuantiG... The Ultimate Topic... Falguni Pratham & AC AtCoder Problems cp-library/impleme... Um_nik World All Bookmarks

Deploy :

Mathematical word problem solver

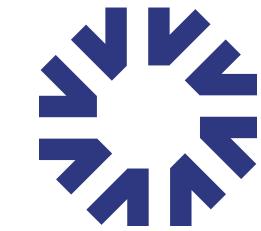
Word Problem

Bryan took a look at I

Press Enter to apply

Predict

The image shows a screenshot of a web browser window. The title bar includes the URL 'localhost:8501' and several bookmarks. The main content area features a large heading 'Mathematical word problem solver'. Below this, a section titled 'Word Problem' contains a text input field with the placeholder 'Bryan took a look at I'. To the right of the input field is a button labeled 'Press Enter to apply'. Below the input field is a button labeled 'Predict'. The background of the page has a faint, abstract pattern.



SPYs

Thank You

29 November, 2024