Learning Multi-Step Reasoning by Solving Arithmetic Task

ACL 2023

Tianduo Wang, Wei Lu

StatNLP Research Group

Singapore University of Technology and Design

Introduction

▪ Math Word Problem Solving

- Answer a mathematical query according to the text description
- Generate equations to obtain final answer

All Problems Require Equation

Math Word Problem Solving ○

Math23K (Wang et al., 2017)

Problem: 1 day, 1 girl was organizing her book case making sure each of the shelves had exactly 9 books on it. She has 2 types of books - mystery books and picture books. If she had 3 shelves of mystery books and 5 shelves of picture books, how many books did she have in total?

Equation & Solution

$(1 \times 1) + (9 \times 1) = x$

$x = 10$

Operators

$+, -, \times, \div$

Problem Type:

Linear Algebra

Some Problems Without Equation

Math Quesition Answering ✗

DROP (Dheeru Dua et al, 2019)

Passage (some parts shortened)

In **1517, the seventeen-year-old King sailed to Castile**. There, his Flemish court .... **In May 1518, Charles traveled to Barcelona in Aragon**.

Question

Where did Charles travel to first, Castile or Barcelona?

Reasoning

1517: *Castile*
1518: *Charles*
1517 > 1518

Solution

Castile

Yan Wang et al. Deep Neural Solver for Math Word Problems. EMNLP. 2017.
Dheeru Dua et al. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. NAACL. 2019.

Introduction

▪ Mathematical Reasoning

- ◦ ★ **Math Word Problem Solving**
- ◦ Theorem Proving
- ◦ Geometry Problem Solving
- ◦ Math Question Answering
- ◦ Other Quantitative Problems
    - Diagram, Finance, Science, Programming

Pan Lu et al. A Survey of Deep Learning for Mathematical Reasoning. 2022.

Introduction

▪ Mathematical Reasoning

- Chain-of-thought prompting (Wei et al., 2022)
  - Pros: Elicit LLM's ability to decompose a complex problem into several intermediate steps
  - Cons: Such ability only emerges from sufficiently large models (↑100B parameters)
- **To address the issue: Our proposed work**
  - Examine how to incorporate moderate-sized LMs
    e.g., RoBERTa (Liu et al., 2019), with such multi-step reasoning ability
    via continual pre-training to improve the performance on math problems

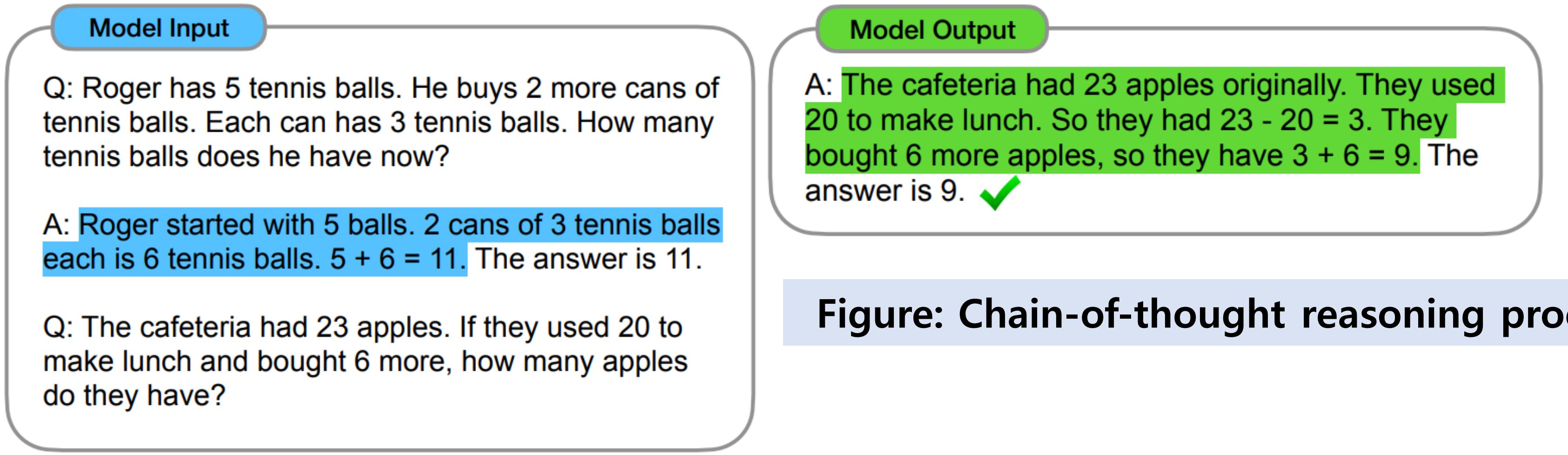


Figure: Chain-of-thought reasoning processes

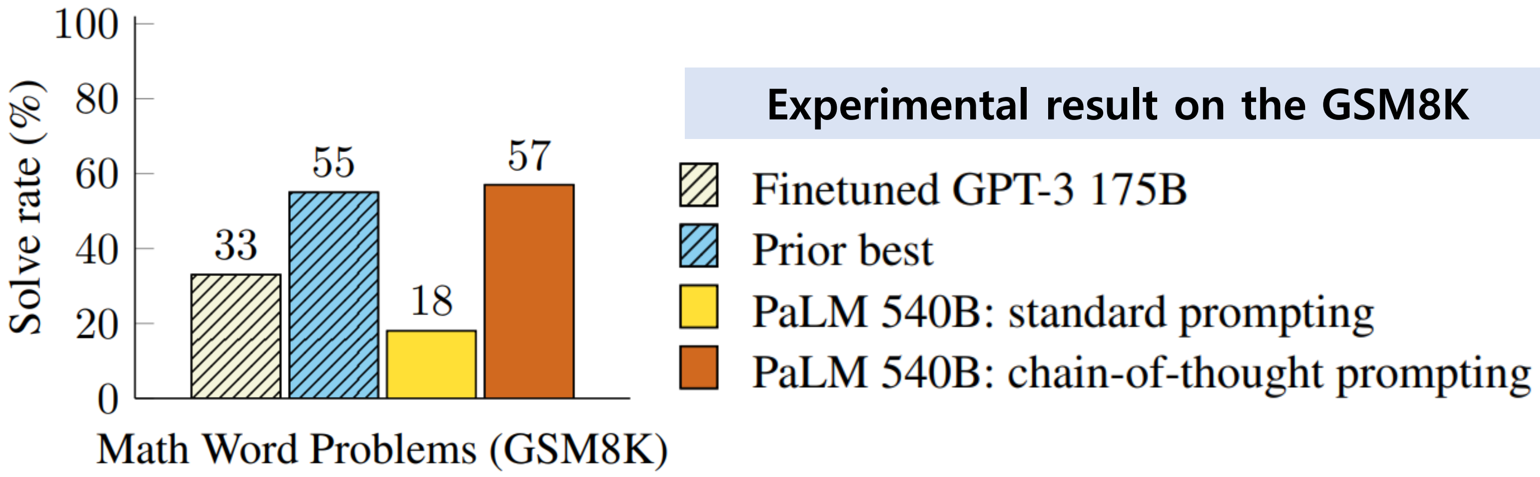
Jason Wei et al. Chain of thought prompting elicits reasoning in large language models. NeurIPS. 2022.
Yinhan Liu et al. RoBERTa: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692. 2019.

Introduction

▪ Mathematical Reasoning

- Chain-of-thought prompting (Wei et al., 2022)
  - Pros: Elicit LLM's ability to decompose a complex problem into several intermediate steps
  - Cons: Such ability only emerges from sufficiently large models (⬆ 100B parameters)
- **To address the issue: Our proposed work**
  - Examine how to incorporate moderate-sized LMs
    e.g., RoBERTa (Liu et al., 2019), with such multi-step reasoning ability
    via continual pre-training to improve the performance on math problems

Experimental result on the GSM8K



Medium-sized LMs

Model	Size
BERT$_{BASE}$	110M
BERT$_{LARGE}$	336M
ROBERT$_{BASE}$	125M
ROBERT$_{LARGE}$	355M

Jason Wei et al. Chain of thought prompting elicits reasoning in large language models. NeurIPS. 2022.
Yinhan Liu et al. RoBERTa: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692. 2019.

Part 1.

Introduction

▪ Correctly understanding numbers

- Medium-sized LMs have a deficiency in numerical comprehension
- **Previous work**
  - Masking numbers with special tokens, and generating symbolic expressions with a structured neural decoder
  - Pre-trains LMs on synthetic numerical tasks, which requires models to learn how to perform computation involving numbers

- **Critical limitations**
  - For symbolic methods, they neglect the information carried by the numbers

➡ As for continual pre-training methods, LMs' arithmetic skills are not reliable
Such skills are highly influenced by the training data and hard for extrapolation

Question: Roger has 5 (<Num0>) tennis balls. He buys 2 (<Num1>) more cans of tennis balls. Each can have 3 (<Num2>) tennis balls. How many tennis balls does he have now?

Math expression: $5 + 2 \times 3$ **Ans:** 11

Symbolic expression: <Num0> + <Num1> × <Num2>

Chain-of-thought (Wei et al., 2022):
Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Eric Wallace et al. Do NLP models know numbers? probing numeracy in embeddings. EMNLP-IJCNLP. 2019.
Jason Wei et al. Chain of thought prompting elicits reasoning in large language models. NeurIPS. 2022.

▪ Proposed methods

- **Training Process**
  - First pre-train moderate-sized LMs on a synthetic dataset called MSAT (Multi-step Arithmetic Tasks)
  - Then fine-tune on downstream task

- To make sure LMs **capture the information carried by the numbers**
  - Keep the numbers in the questions instead of masking them during both pre-training and finetuning

- **Instead of making LMs conduct computation internally**
  - MSAT encourages LMs to generate a series of intermediate steps leading to the answer

Method

▪ Digit tokenization for numbers

- **Sub-word tokenization methods, e.g., Byte Pair Encoding (BPE)**
  - BPE-based tokenizers split text based on the token frequency in the training corpus, which can be counter-intuitive when dealing with numbers
  - **Such inconsistent tokenization strategy** for numbers undermines LM’s numerical understanding ability

- **To address the issue**
  - Tokenize numbers digit-by-digit for both pre-training and fine-tuning

Sub-word Tokenization

Number			Tokenized Number	
“520"	"521"	➡	["520"]	["5", "21"]

Digit-by-digit Tokenization

Number			Tokenized Number	
“520"	"521"	➡	["5”, “2”, “0"]	["5", "2, “1"]

Rico Sennrich et al. Neural machine translation of rare words with subword units. ACL. 2016.
Eric Wallace et al. Do NLP models know numbers? probing numeracy in embeddings. EMNLP-IJCNLP. 2019.

Method

▪ Multi-step Arithmetic Tasks (MSAT)

◦ Seq2Seq task

- **Input**: a question context, equation, and question variable
  - **Equation** is a sequence of symbols and operators (+, −, ×, ÷, =) that builds equality relationship between symbols
  - Only one of the symbols is set as the **question variable**, while other symbols will be listed in **question context** with their numerical values

- **Output**: **a multi-step reasoning chain** leading to the answer
  - Each step consists of two sub-steps: variable assignment and calculation
  - **Variable assignment**: Numbers appear in the input sequence are assigned to the variable names that are exclusive for decoder
  - **Calculation:** a new variable is generated from the calculation of the existing variables

Question: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can have 3 tennis balls. How many tennis balls does he have now?
<Num0> <Num1> <Num2>

Math expression: $5 + 2 \times 3$ **Ans:** 11

Symbolic expression: <Num0> + <Num1> × <Num2>

Code-style multi-step expression (ours):

```
N0 = 2. N1 = 3. N2 = N1 * N0.     (step 1)
N3 = 5. Ans = N2 + N3.            (step 2)
```

Method

▪ Construction of MSAT (Multi-step Arithmetic Tasks)

◦ **Input sequence construction**

- Each equation template contains no more than **3 binary operators** (+, −, ×, ÷)
- Instantiate an equation from an equation template
  - **An equation template** "`<Num0> + <Num1> = <Num2>`"
  - Assign each variable a value that makes the equality hold and a variable name selected from the capitalized letters
  - The numbers in the questions are sampled from 0 to 10,000
- Randomly pick a variable as the **question variable**

- The resulting input arithmetic question "`A=1. C=3. A+B=C. B?`"

MSAT Example

Item	Text
Question	B = number0 . B + number1 = N . N ?
Equation	+ number0 number1
Numbers	343 32
Answer	375

Decompose Question Context

◦ B = number0
◦ B + number 1 = N
◦ N?

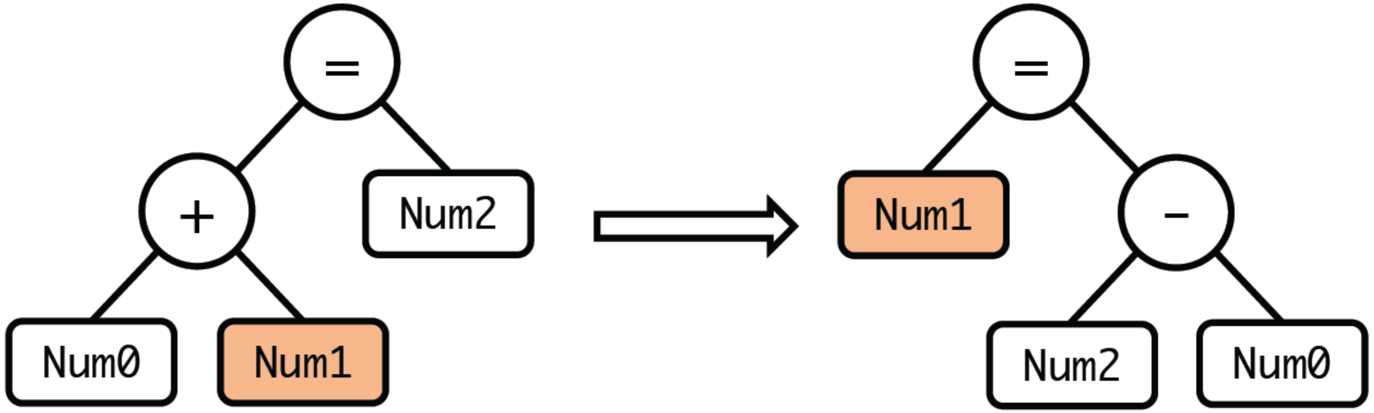
Method

▪ Construction of MSAT (Multi-step Arithmetic Tasks)

◦ Output sequence construction

- Given an equation and a question variable, the output is first constructed as a math expression leading to the value of the question variable
- Equation can be represented as a binary tree
  - The variables are the terminal nodes and operators are the non-terminal nodes

Tree inversion algorithm



An arithmetic question

An output expression

***Question variable is highlighted**

Method

▪ Pre-training via adapter-tuning

- **Directly training on synthetic data harms performance (Geva et al., 2020).**
  - Synthetic data are largely different from the natural language corpus

- **Adopt a two-stage tuning strategy (Wang and Lu, 2022):**
  1) Perform adapter-tuning (Houlsby et al., 2019) on MSAT
  2) Jointly fine-tune adapter and LM backbone on downstream tasks

  - **Inject reasoning skills into model**
  - **It mitigates catastrophic forgetting**
    because LM's original parameters are largely preserved during adapter-tuning

Mor Geva et al. Injecting numerical reasoning skills into language models. ACL. 2020.
Xuezhi Wang et al. Self-consistency improves chain of thought reasoning in language models. ICLR. 2023.
Neil Houlsby et al. Parameter-efficient transfer learning for nlp. ICML. 2019.

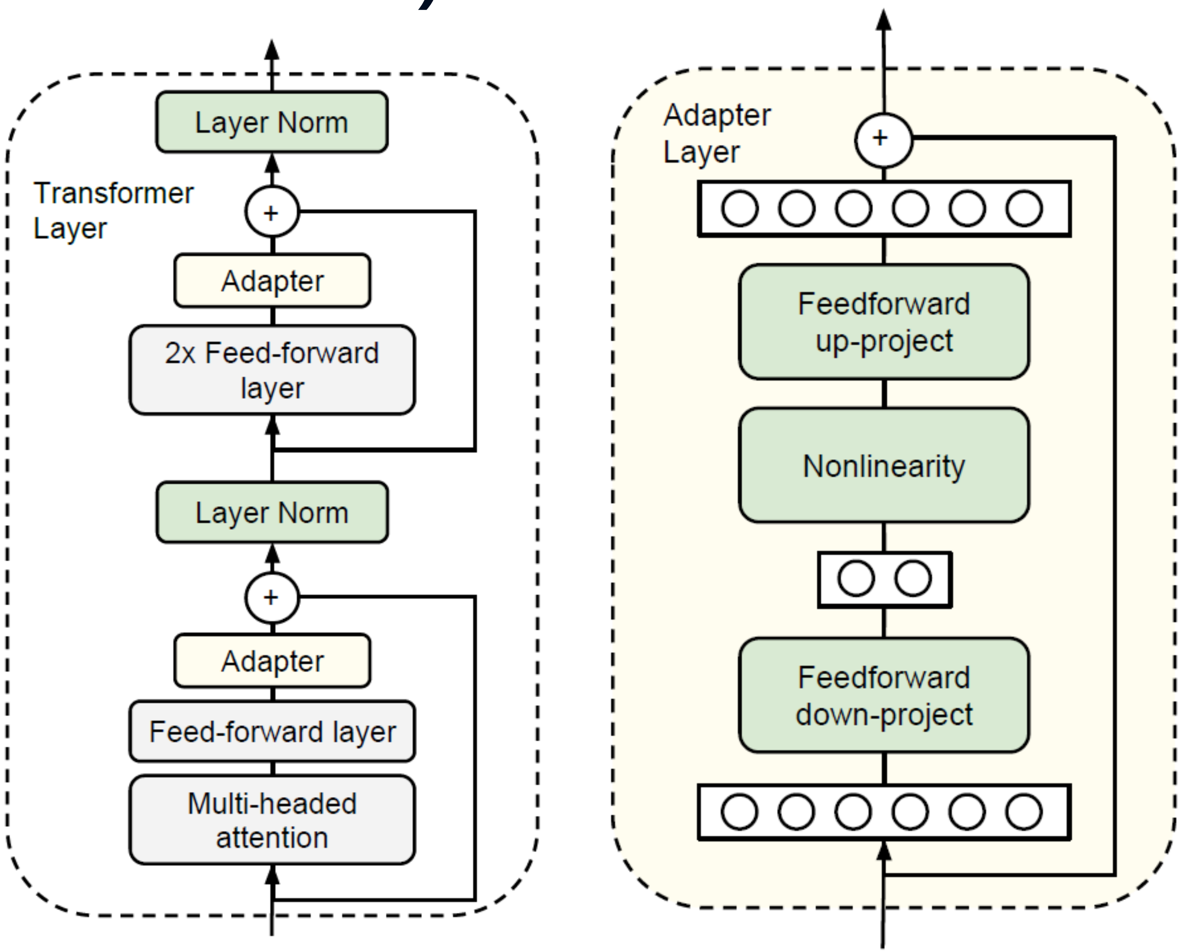
Method

- **Pre-training via adapter-tuning**
  - Consider two backbone models to verify the effectiveness of our method
    - **Select models for adopting** $\text{RoBERTa}_{\text{base}}$ to encode the input questions
      - A sequence-to-sequence (RoBERTaGEN) model (Lan et al., 2021)
      - Directed acyclic graph (DAG) structured model (Jie et al., 2022)

Yihuai Lan et al. Mwptoolkit: An open-source framework for deep learning-based math word problem solvers. arXiv preprint arXiv:2109.00799. 2021.
Zhanming Jie et al. Learning to reason deductively: Math word problem solving as complex relation extraction. ACL. 2022.

Method

▪ Two-stage tuning strategy: Adapter-tuning (Houlsby et al., 2019)



- **Vanilla fine-tuning**
  - A modification is made to the top layer of the network
- **Tuning with adapter modules**
  - The weights of the original network are untouched whilst the new adapter layers are initialized at random
  - Train new layer normalization parameters per task

Result on GLUE (a collection of 9 NLP tasks)

	Total num params	Trained params / task	Total
$\text{BERT}_{\text{LARGE}}$	9.0×	100%	80.4
Adapters (8-256)	1.3×	3.6%	80.0
Adapters (64)	1.2×	2.1%	79.6

Neil Houlsby et al. Parameter-efficient transfer learning for nlp. ICML. 2019.

Method

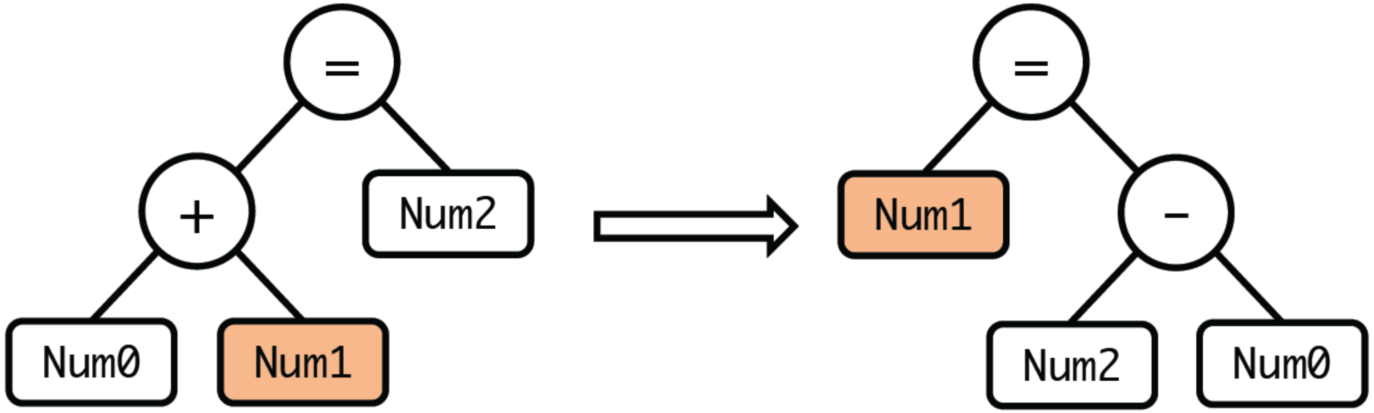
- **Pre-training via adapter-tuning**
  - Consider two backbone models to verify the effectiveness of our method
    - **Select models for adopting** $RoBERTa_{base}$ to encode the input questions
      - A sequence-to-sequence ($RoBERTa_{GEN}$) model (Lan et al., 2021)
      - Directed acyclic graph (DAG) structured model (Jie et al., 2022)

Yihuai Lan et al. Mwptoolkit: An open-source framework for deep learning-based math word problem solvers. arXiv preprint arXiv:2109.00799. 2021.
Zhanming Jie et al. Learning to reason deductively: Math word problem solving as complex relation extraction. ACL. 2022.

Method

- **Two-stage tuning strategy: Jointly fine-tune adapter and LM backbone on downstream tasks**
  - **Output sequence construction**
    - Given an equation and a question variable, the output is first constructed as a math expression leading to the value of the question variable
    - Equation can be represented as a binary tree
      - The variables are the terminal nodes and operators are the non-terminal nodes

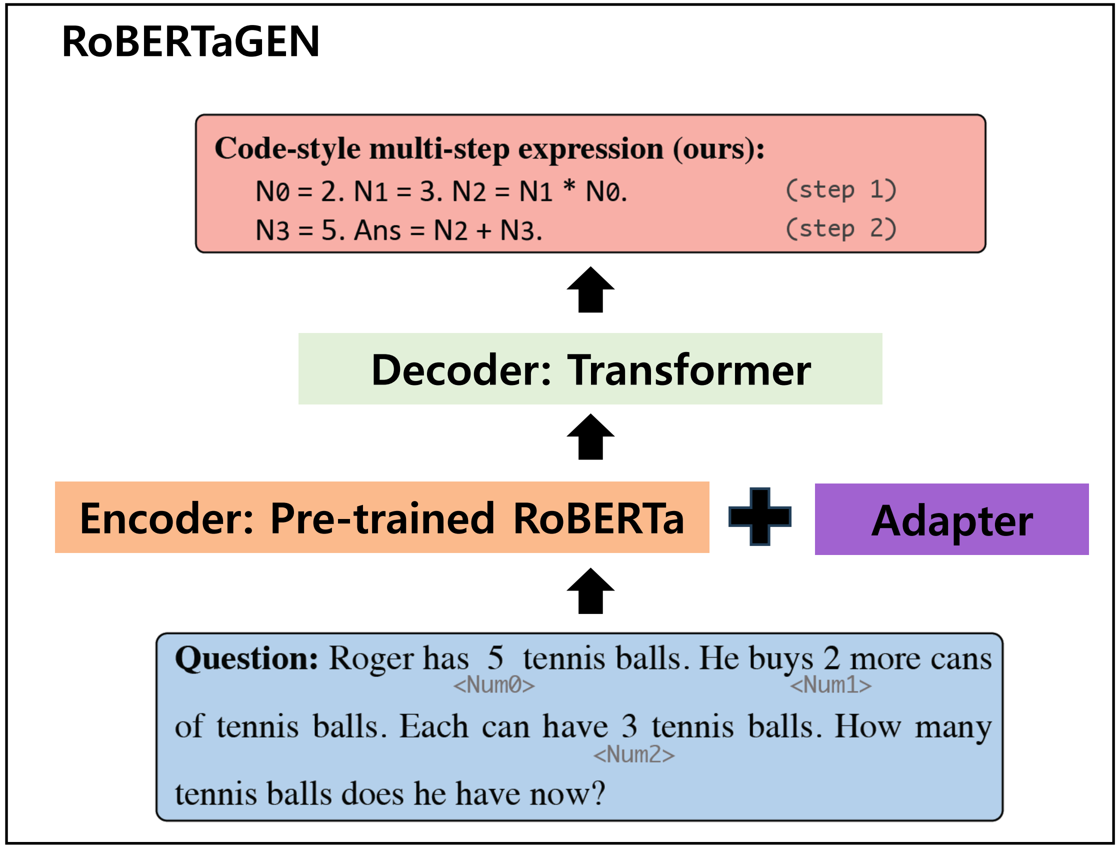
Tree inversion algorithm



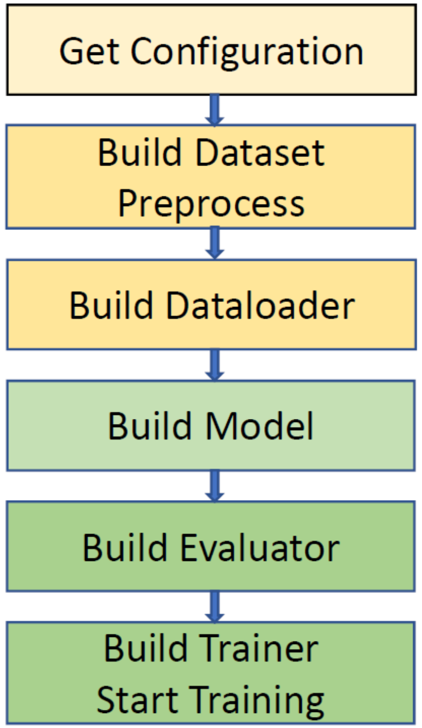
Yihuai Lan et al. Mwptoolkit: An open-source framework for deep learning-based math word problem solvers. arXiv preprint arXiv:2109.00799. 2021.

Method

- **Backbone: Sequence-to-sequence model (RoBERTaGEN))**
  - **Run RoBERTaGEN in MWPToolkit (Lan et al., 2021)**



MWPToolkit



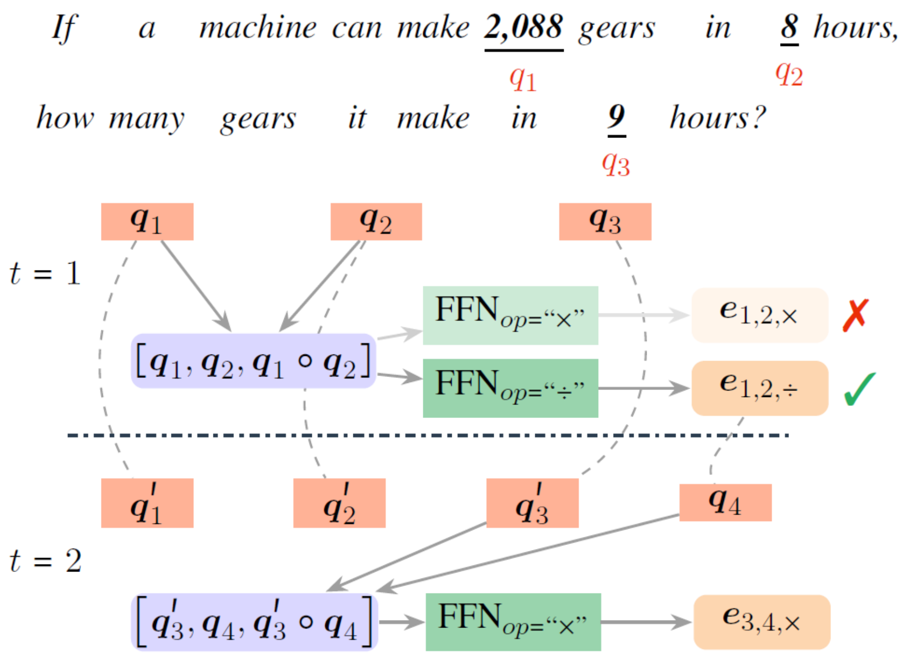
Open-source framework for deep learning-based MWP solvers

- Users can run, compare, and test models with simple configuration files and command lines

Yihuai Lan et al. Mwptoolkit: An open-source framework for deep learning-based math word problem solvers. arXiv preprint arXiv:2109.00799. 2021.

Part 2. Method

- **Backbone: Directed acyclic graph structured model (Jie et al., 2022)**
  - **Observation: MWP solving can be viewed as a complex relation extraction problem**
    - The task of identifying the complex relations among the quantities in the problem text
    - Each primitive arithmetic operation ("+", " − ", ...) defines a different type of relation



Deductive Reasoner

- **The procedure to obtain the expression** “$q_1 \div q_2 \times q_3$”
  - Encoding: e.g., 2,088 -> a quantity token “ $< quant >$ "
  - Adopt a pre-trained language model ROBERTa
  - Obtain the representation of quantity pairs (q_i, q_j) at step t
  - Assign a score for the best expression
  - Decide termination step of the procedure

Zhanming Jie et al. Learning to reason deductively: Math word problem solving as complex relation extraction. ACL. 2022.

Experiments

▪ Benchmark

- MAWPS
- ASDiv-A
- SVAMP
  - Variations(changing questions, adding irrelevant information, etc) from ASDiv-A
- SVAMP (hard)
  - Evaluate models' extrapolation ability on the out-of-distribution numbers
  - Replace the original numbers in SVAMP

Existing dataset statistics

Dataset	# Data	Avg. input length	Avg. output reasoning steps
MAWPS	1,987	30.3	1.4
ASDiv-A	1,217	32.3	1.2
SVAMP	1,000	34.7	1.2

Experiments

▪ Baselines

- Seq2Seq model(139.71M)
  - ROBERTaGEN(w/ or w/o MSAT) + Transformer decoder
- DAG structured model(142.40M)
  - DeductReasoner(w/ or w/o MSAT) + DAG decoder
- LLMs
  - PaLM(540B), Codex(175B), Chain-Of-Thought prompting (COT)

▪ Tokenization

- Previous work: Symbolic mask tokens
  - Models(Seq2Seq model, DAG structured model) replace numbers with symbolic mask tokens
- Ablation study on our work: Digit tokenization
  - Both models uses actual numbers with digit tokenization

Experiments

▪ Baselines

- Digit tokenization baselines perform worse than symbolic mask counterparts
- The models trained with MSAT surpass both baselines
- SVAMP (hard): Our model is more robust in handling out-of-distribution numbers

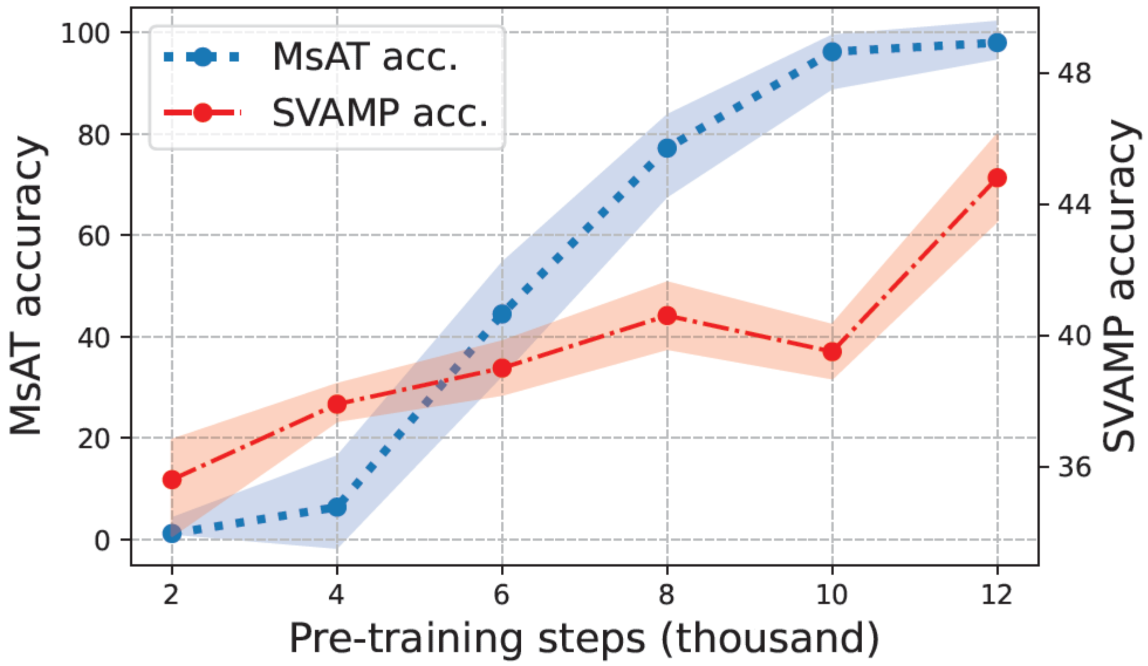
Model	MAWPS		ASDiv-A		SVAMP		SVAMP (hard)	
	Acc.	Δ	Acc.	Δ	Acc.	Δ	Acc.	Δ
Large language models	(PaLM 540B)		(code-davici-002)		(PaLM 540B)			
w/ Chain-of-Thought prompting	93.3		80.4		**79.0**		-	
Seq2Seq models								
RoBERTaGen (Lan et al., 2021) (139.71 M)								
w/ symbolic masks	88.4		72.1		30.3		30.3♡	
w/ digit tokenization	84.1	(-4.3)	71.9	(-0.2)	27.6	(-2.7)	19.6	(-10.7)
MsAT-RoBERTaGen (Ours)	**91.6**	(+3.2)	**81.8**	(+9.7)	**39.8**	(+9.5)	**36.2**	(+5.9)
DAG structured models								
DeductReasoner (Jie et al., 2022) (142.40M)								
w/ symbolic masks	92.0		85.0		45.0		45.0♡	
w/ digit tokenization	91.6	(-0.4)	84.1	(-0.9)	44.4	(-0.6)	42.8	(-2.2)
MsAT-DeductReasoner (Ours)	**94.3**	(+2.3)	**87.5**	(+2.5)	**48.9**	(+3.9)	**48.2**	(+3.2)

Experiments

▪ Pre-training analysis

- Learn multi-step reasoning gradually from the synthetic task MSAT
- Pre-training task MSAT is transferred to the downstream MWP solving tasks

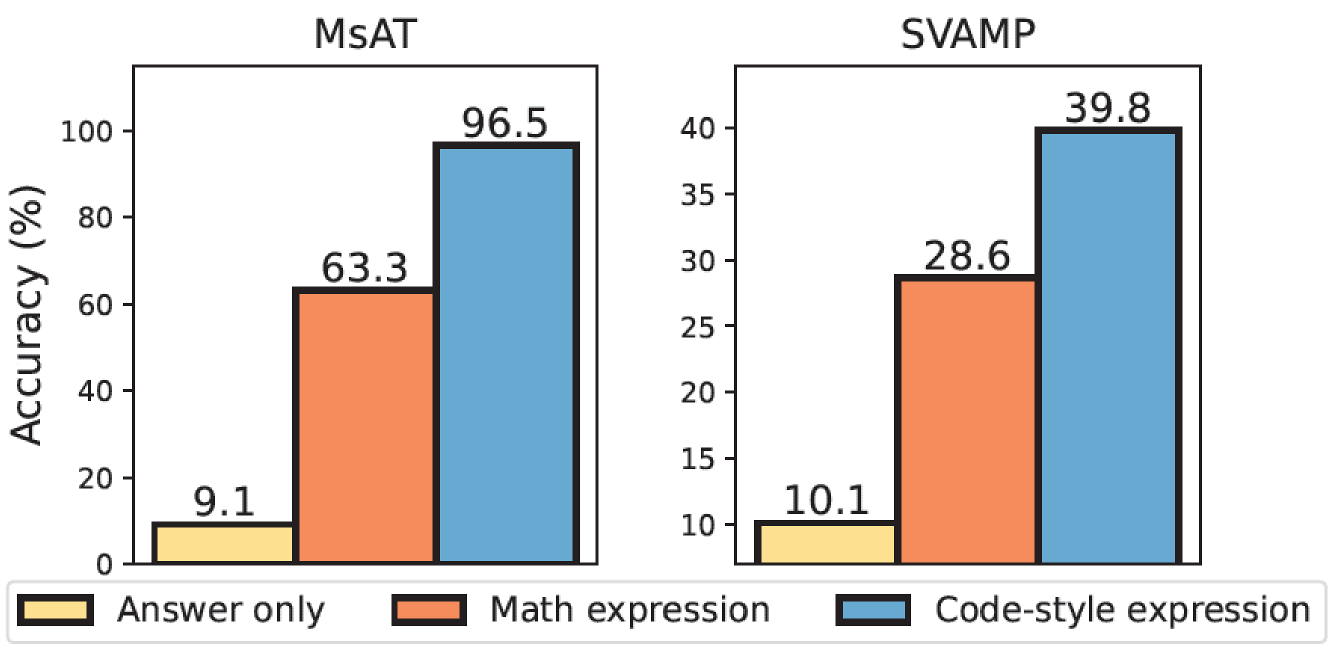
Result with expect to the pre-training steps



▪ Reasoning format of MSAT

- Substitute MSAT for step-by-step output sequences with only numerical answers
- Confirm the necessity of producing intermediate reasoning steps during pre-training

Comparison between different output expression formats

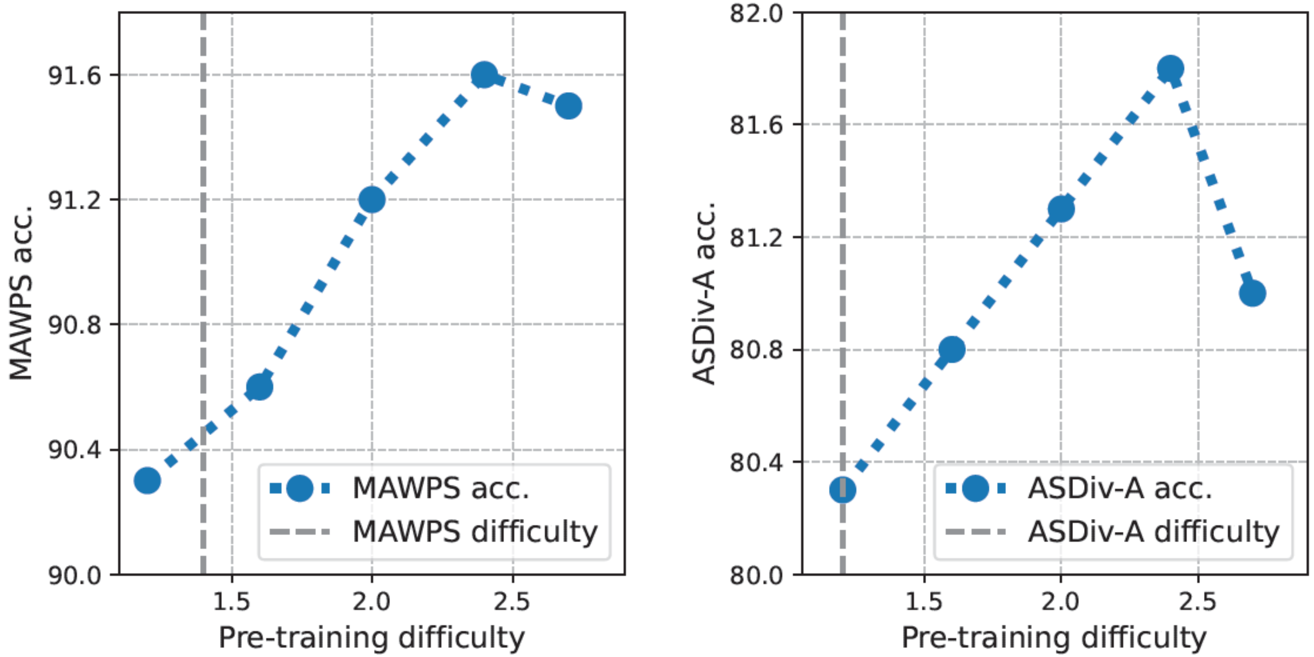


Experiments

▪ Difficulty level of MSAT

- The advantage of enabling highly customizable difficulty levels for the training data
- The difficulty level of a reasoning task is not solely determined by the number of reasoning steps

Performance with respect to pre-training difficulty

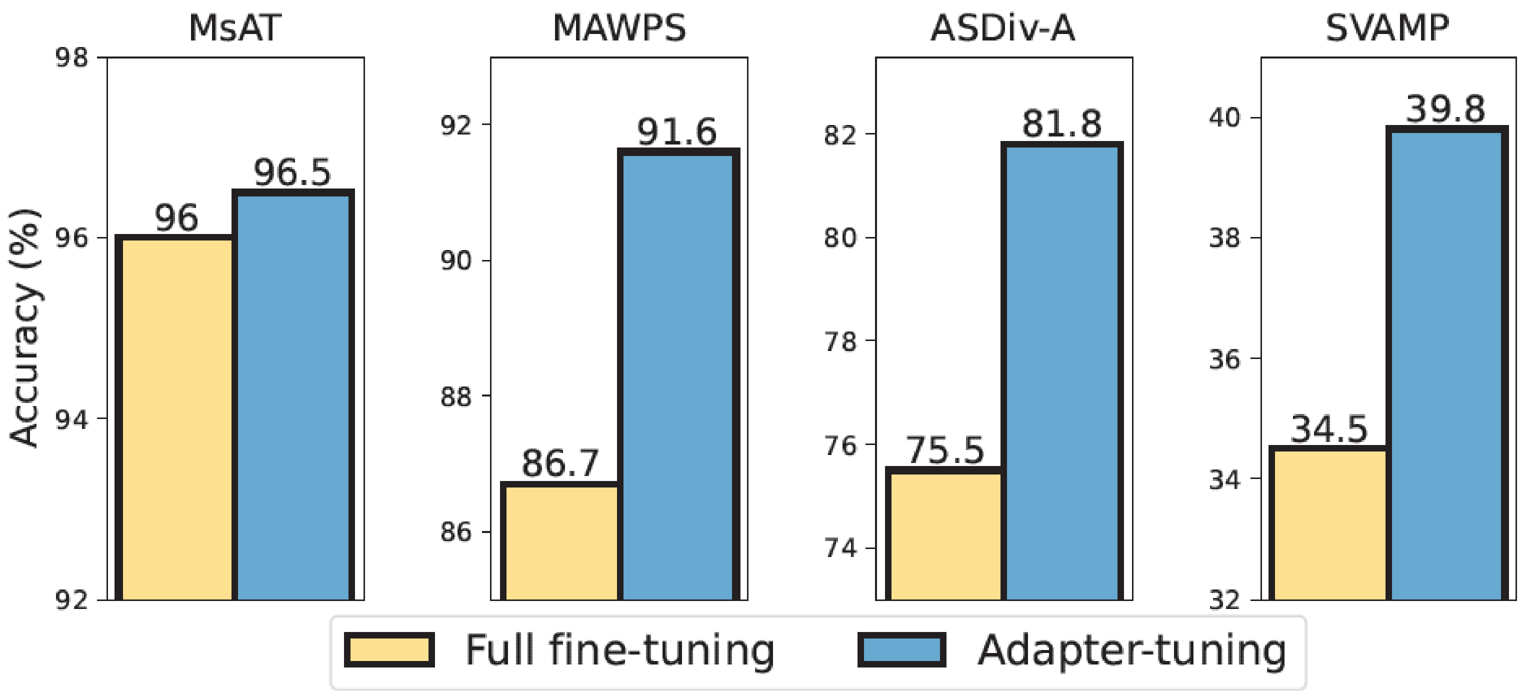


Experiments

▪ Perform adapter-tuning on MSAT

- Adapter-tuning outperforms fine-tuning on all downstream MWP datasets
- Demonstrate the benefits of performing adapter-tuning on MSAT

Result according to the pre-training steps



Conclusion

- **Synthetic pre-training task, MSAT**
  - Incorporate LMs with multi-step reasoning skills that improve performance on MWP tasks
  - Encourage LMs to generate intermediate reasoning steps instead of predicting final numerical answers directly
  - The proposed method is effective in improving the moderate-sized LM's performance on MWP solving tasks

Conclusion

- **Limited number of operators considered**
  - Previous methods
    - Only consider binary operators (+, −, ×, and ÷)
  - Adopt a code-style output format
    - Introduce other non-binary operators supported by the Python interpreter
      e.g., $sum()$ and $\max()$
  - Obtain labeled data with such operators may require laborious efforts
  - It is an interesting research question on exploring how to teach models to solve practical questions
    - e.g., math word problems, by writing code in a low-resource setting (Jie and Lu, 2023)