A Cognitive Solver with Autonomously Knowledge Learning for Reasoning Mathematical Answers

Jiayu Liu[1,2], Zhenya Huang[1,2*], Xin Lin[1,2], Qi Liu[1,2], Jianhui Ma[1,2], Enhong Chen[1,2]

[1]Anhui Province Key Laboratory of Big Data Analysis and Application, School of Data Science & School of Computer Science and Technology, University of Science and Technology of China (USTC);

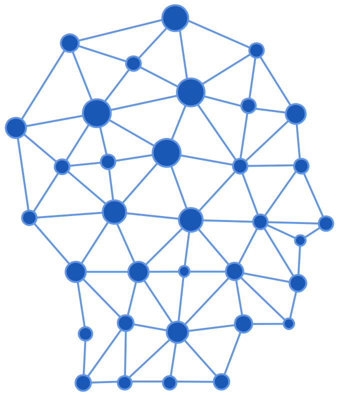
[2] State Key Laboratory of Cognitive Intelligence

Reporter: Jiayu Liu

ICDM-2022

中国科学技术大学
University of Science and Technology of China

Outline



Background

Automatically solving mathematical problems

- Crucial step towards general artificial intelligence
- Requirements:
  - Learn knowledge from data
  - Understand mathematical logics
  - Conduct cognitive reasoning like a human
- The ability to reason mathematical answers is a sign of the level AI achieves

Math word problem	Jack has 3 apples and Amy has 2 bananas, how many fruits do they have?
Expression: 3+2	**Answer:** 5

Background

Problem definition of MWP

- Input: a sequence of n words and numeric values $P = \{p_1, p_2, \dots, p_n\}$
  - E.g., "Jack has 3 apples … "
- Output: mathematical expression E_P, answer S_P
  - $E_P = \{y_1, y_2, \dots, y_m\}$, where y_i comes from $V_P = V_O \cup V_C \cup N_P$
    - ✓ V_O: operators, e.g. $\{+, \times, -, \div\}$
    - ✓ V_C: numeric constants, e.g. $\{1, \pi\}$
    - ✓ N_P: numeric variables from P, e.g. $\{3,2\}$
    - ✓ E.g., "3+2"
  - S_P: real value
    - ✓ E.g., 5

Math word problem	Jack has 3 apples and Amy has 2 bananas, how many fruits do they have?		
Expression:	3+2	**Answer:**	5

Background

Related Work

- Traditional work
  - Rule-based Methods
  - Statistic-based Methods
  - Semantics parsing-based Methods

  Tremendous human effort and low generality
- DL-based Methods
  - Seq2Seq framework
    - ✓ Inspired from language translation research
  - Advanced: Seq2Tree, Graph2Tree, Seq2DAG…

Background

Certain gap from human-like AI

1. **Humans learn knowledge from solving mathematical problems**
   - E.g., “banana” is a kind of “fruit”
   - Interpretable to humans and can be expressed explicitly

- Existing methods: train models to simulate comprehension of problems
  - E.g., better understand the semantic meaning and sentence structure
  - Learning results are often represented as neural networks

Math word problem	Jack has 3 apples and Amy has 2 bananas, how many fruits do they have?
Expression: 3+2	**Answer:** 5

Background

Certain gap from human-like AI

2. Humans are able to apply learned knowledge to answer unseen problems

- E.g., learn that "banana" is a kind of "fruit" from solving the problem, and apply to "<u>John has 2 bananas and Lisa has 4 pears, how many fruits do they have</u>?".
- Important to build an AI with high generalization performance

Our goal: empower machines with the ability to autonomously learn and apply knowledge as humans

Background

Insights: Dual process theory

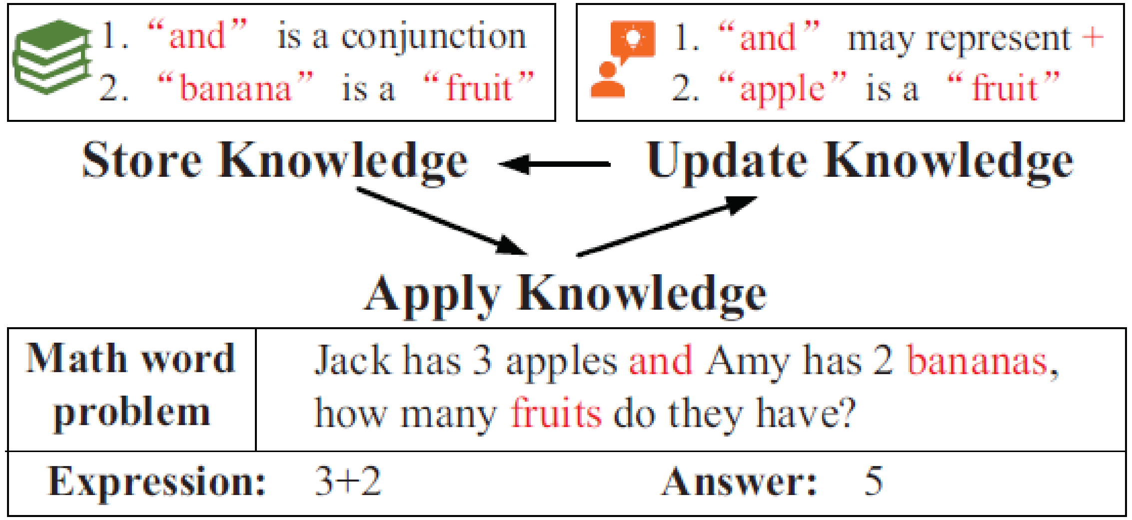
- For cognitive process, two systems: System 1 and System 2
  - System 1 : retrieve relevant information via a fast, unconscious and instructive process
  - System 2 : conduct deeper cognitive reasoning in an analytic and sequential manner

- We establish two systems: *BRAIN-ARM*
  - *BRAIN*: fetches and provides information related to a math word problem
  - *ARM*: does reasoning and generates the solution step by step

Background

Insights: Information processing theory

- learning process could be summarized as *Store-Apply-Update* steps iteratively
  - Step 1: store knowledge
  - Step 2: when faced with a MWP, apply the knowledge and conduct reasoning to solve it
  - Step 3: after solving the problem, summarize the experience and update existing knowledge



BRAIN

ARM

Background

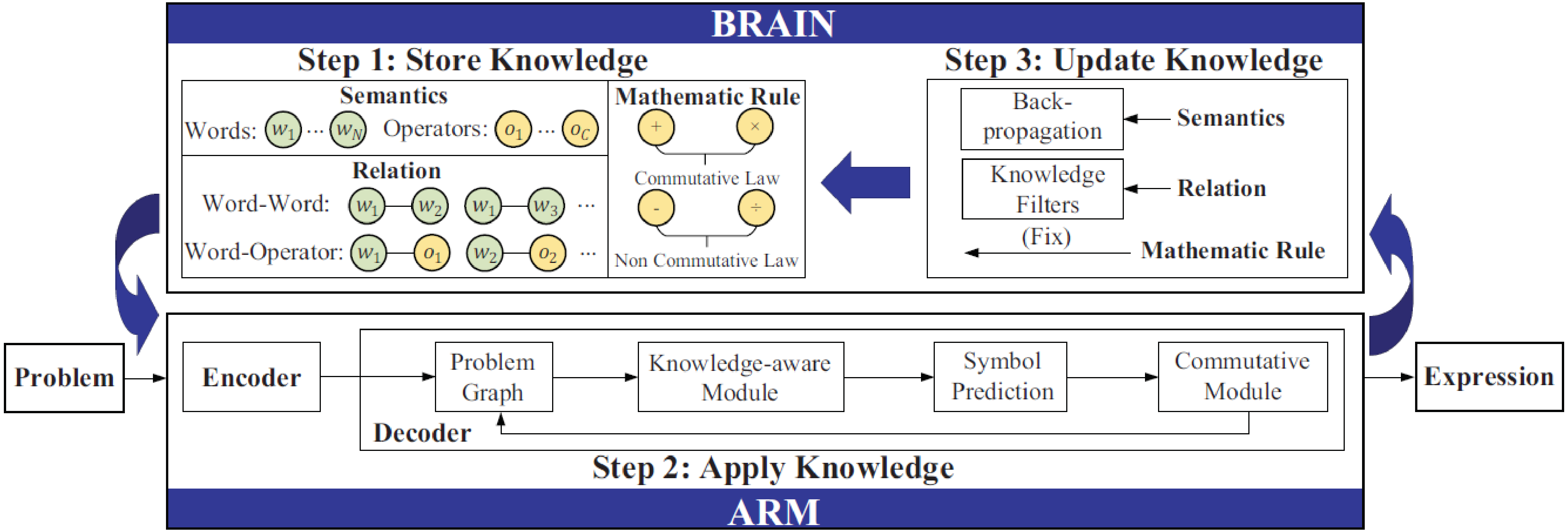
Challenges

- How to storage ?
  - Various types and forms of knowledge
    - ✓ Semantics of tokens, relationships between tokens (e.g. “fruit”—“apple”, “and” —“+”), Mathematical properties (e.g., commutative law)
- How to apply ?
  - We need to design targeted mechanisms to apply different types of knowledge
  - Contextual information should be combined
    - ✓ “and” may represent “+” under certain context, or is just a conjunction
- How to update ?
  - The mechanism of knowledge updating in human’s brain is still not fully understood
  - Coupled with the manner of knowledge storage

Outline

Our Method

CogSolver: Overview



Autonomously learning process:

1. Given a problem, *BRAIN* first retrieves stored knowledge to *ARM*
2. *ARM* applies the knowledge and conducts cognitive reasoning to figure out the problem
3. *BRAIN* updates the knowledge for future application

Our Method

CogSolver: Knowledge storage

- Three types of knowledge
  1. Semantics knowledge
  - **Motivation**: The meanings of mathematical tokens (words and operators) could be represented as feature vectors in the conceptual space
  - **Definition**
    - ✓ Words: $W = \{w_i, i = 1,2, \dots N\}$
    - ✓ Operators: $O = \{o_c, c = 1,2, \dots C\}$

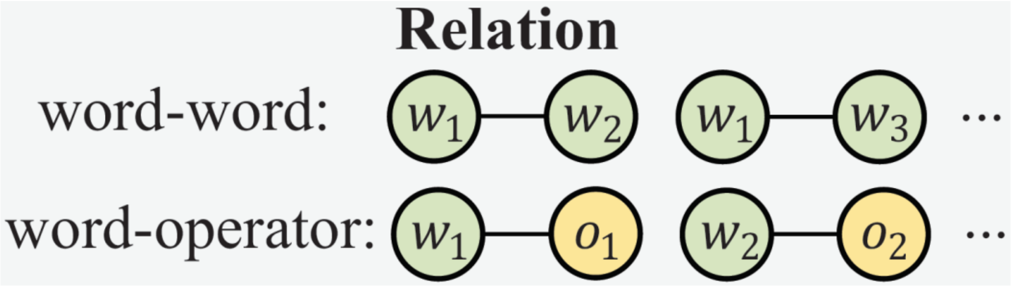
Our Method

CogSolver: Knowledge storage

- Three types of knowledge

  2. Relation knowledge

  - **Motivation**: Relationships between tokens, divided into two types:
    - ✓ Word-word relation (or common-sense)
    - ✓ Word-operator relation
  - **Definition**
    - ✓ word i, j: $ww_{i,j} \in [0,1]$
    - ✓ word i, operator c: $wo_{i,c} \in [0,1]$



  - **Semantics and Relation form a knowledge graph *BG* in *BRAIN***

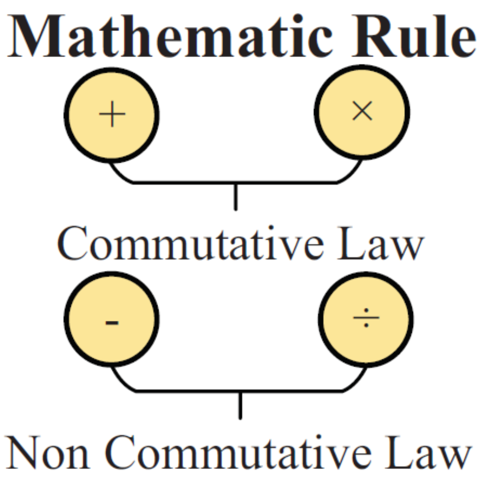
Our Method

CogSolver: Knowledge storage

- Three types of knowledge

  3. Mathematatic rule knowledge

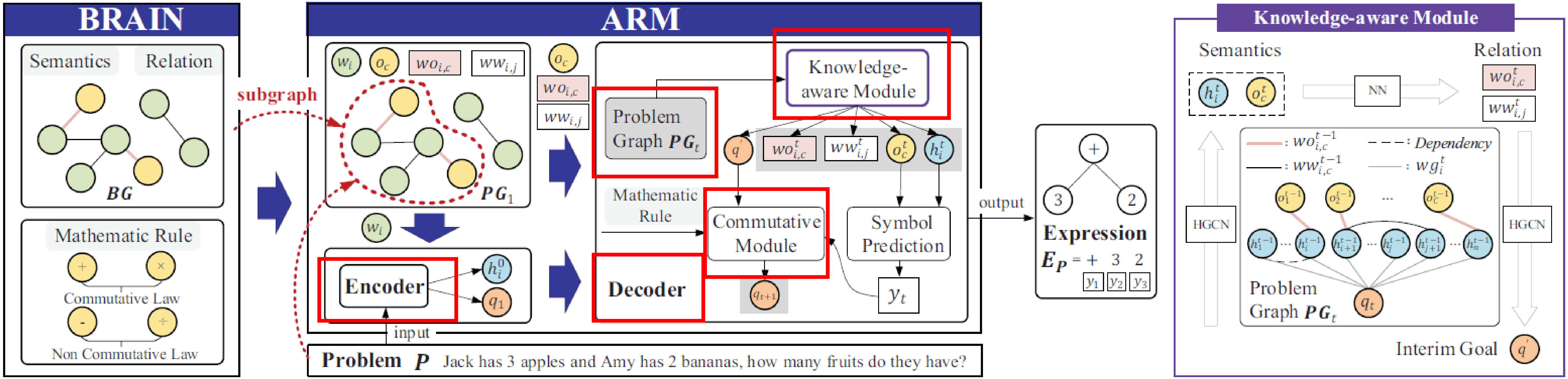
  - **Motivation**: Commutative law is a defined property of operators, a teacher introducing commutative law directly to students



Our Method

CogSolver: Knowledge application

- Problem Solving Process
  - Step 1: Encoder (Semantics knowledge)
  - Step 2: Decoder (Relation, Mathematic rule knowledge)

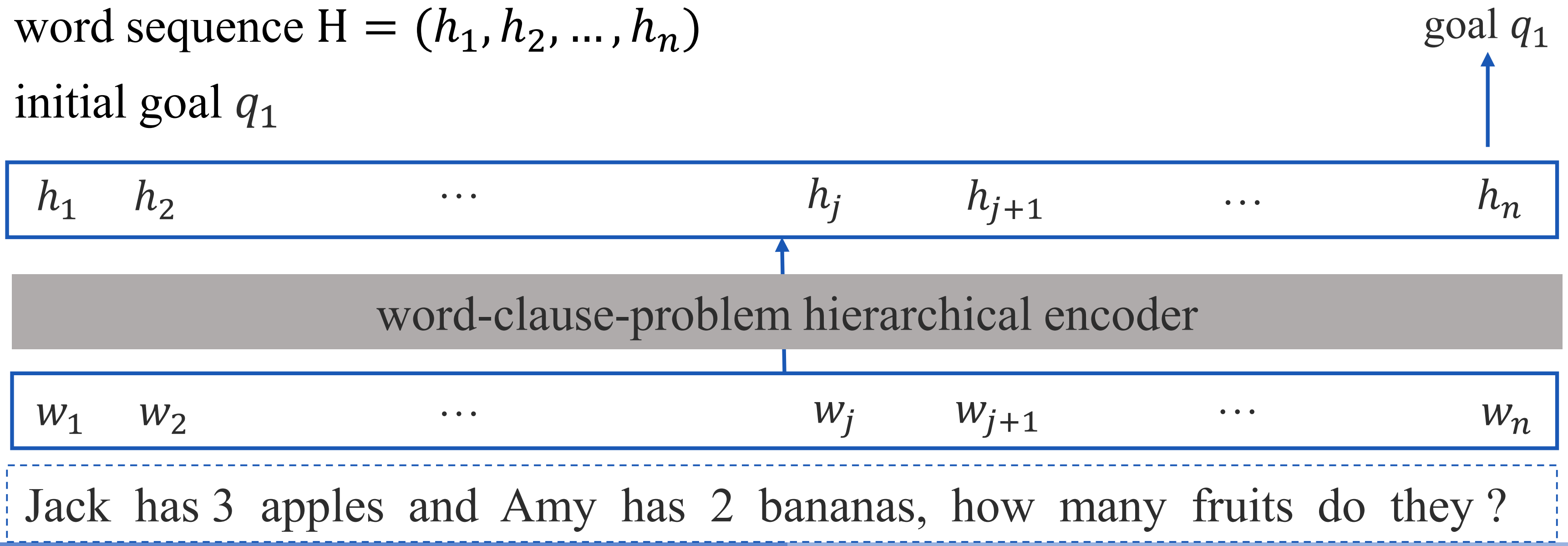


Problem: Jack has 3 apples and Amy has 2 bananas, how many fruits do they have?

Our Method

CogSolver: Knowledge application

- **Encoder**
  - Retrieve Semantics knowledge from *BRAIN* (semantic vectors $w_1, w_2, \dots, w_n$ of words in P)
  - Adopt a word-clause-problem hierarchical encoder
  - Output:
    - ✓ word sequence $\mathrm{H} = (h_1, h_2, \dots, h_n)$
    - ✓ initial goal q_1

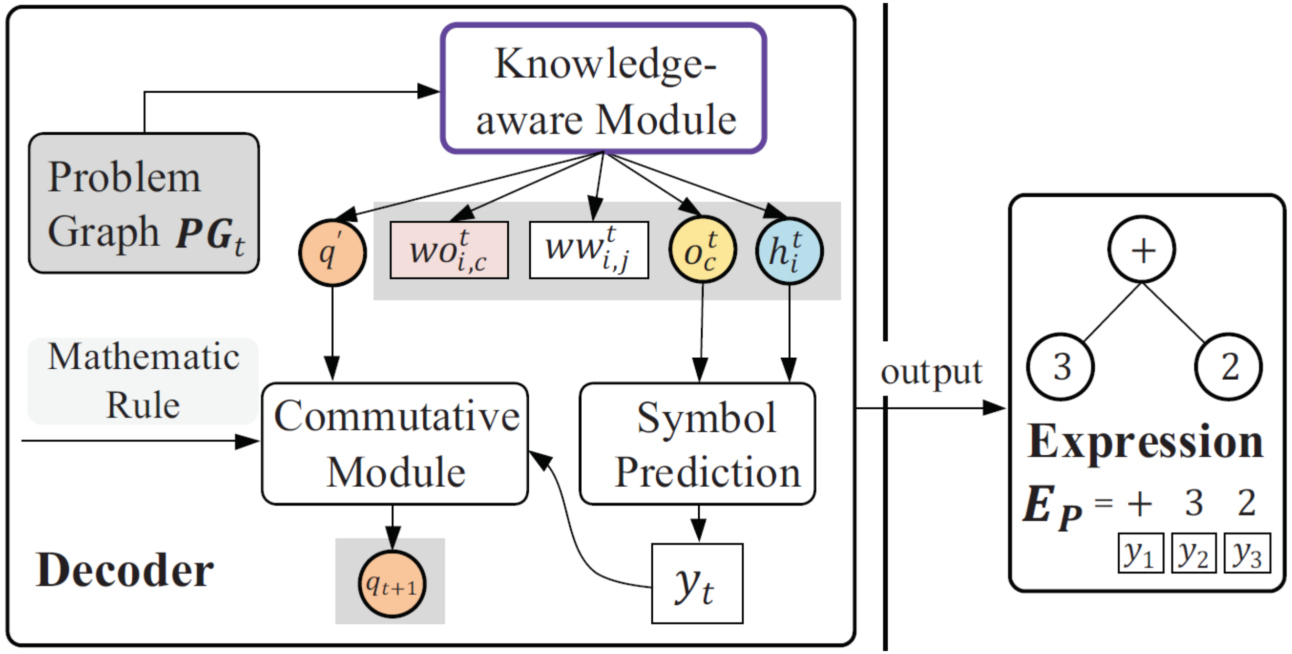


Our Method

Problem: Jack has 3 apples and Amy has 2 bananas, how many fruits do they have?

CogSolver: Knowledge application

- **Decoder:** follows the goal-driven mechanism
  - Contains 4 main parts: Problem Graph, Knowledge-aware Module, Symbol Prediction and Commutative Module

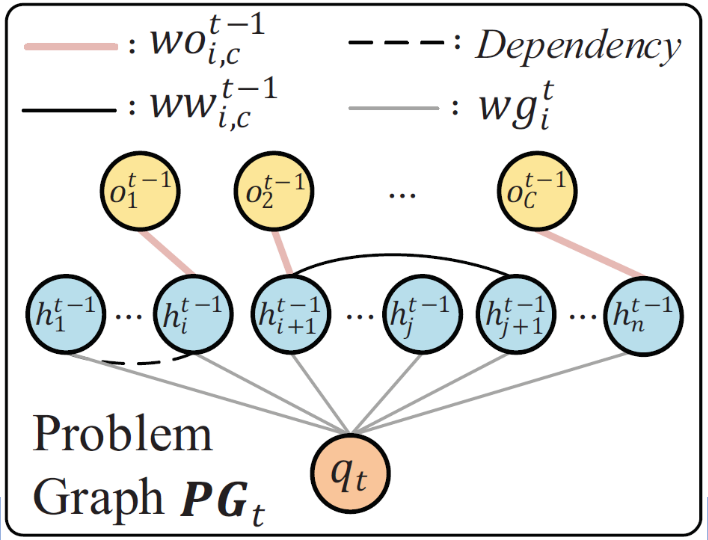


Problem: Jack has 3 apples and Amy has 2 bananas, how many fruits do they have?

Our Method

CogSolver: Knowledge application

- **Decoder**
  - Problem graph $\boldsymbol{PG_t}$: organize semantics, relation knowledge, and goal q_t
    - Node: goal q_t, word h_i^{t-1}, operator o_c^{t-1}
    - Edge: weighted, undirected
      - ✓ words and words, words and operators: $(ww_{i,j}^{t-1}, wo_{i,c}^{t-1})$
      - ✓ Dependency relation in a clause
      - ✓ word-goal wg_i^t (from HMS)
    - $o_c^0, ww_{i,j}^0, wo_{i,c}^0$ are retrieved from BRAIN
    - $\boldsymbol{PG_t}$ is a sub-graph of $\boldsymbol{BG}$ in *BRAIN*



Our Method

Problem: Jack has 3 apples and Amy has 2 bananas, how many fruits do they have?

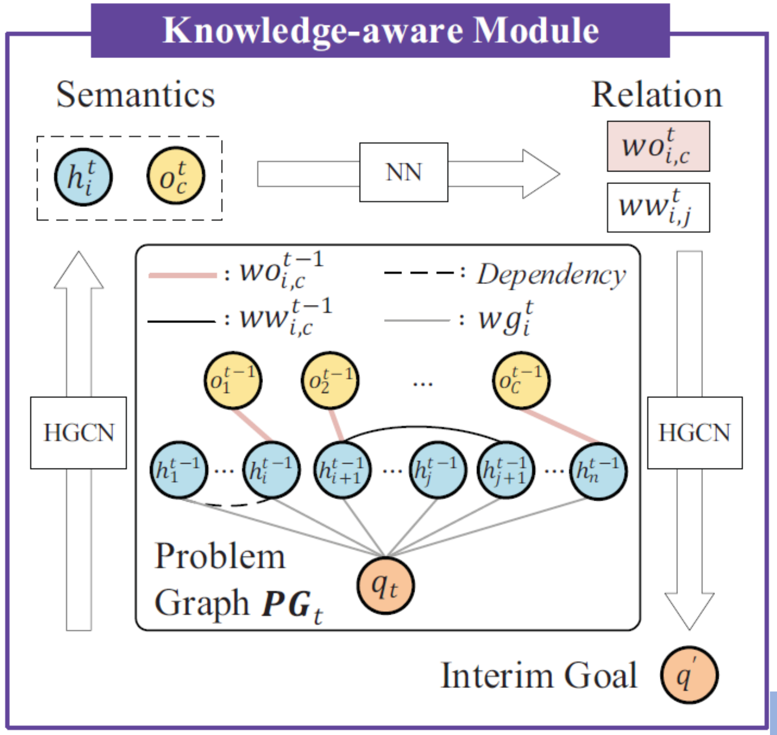
CogSolver: Knowledge application

- **Decoder**
  - Knowledge-aware module
    - ✓ **Motivation:** knowledge may not be suitable to current reasoning step
    - ✓ E.g., “and” is related to + ?

+ - × / *

Jack … apple and Amy … bananas, how... they have?

goal: How many fruits do they have?

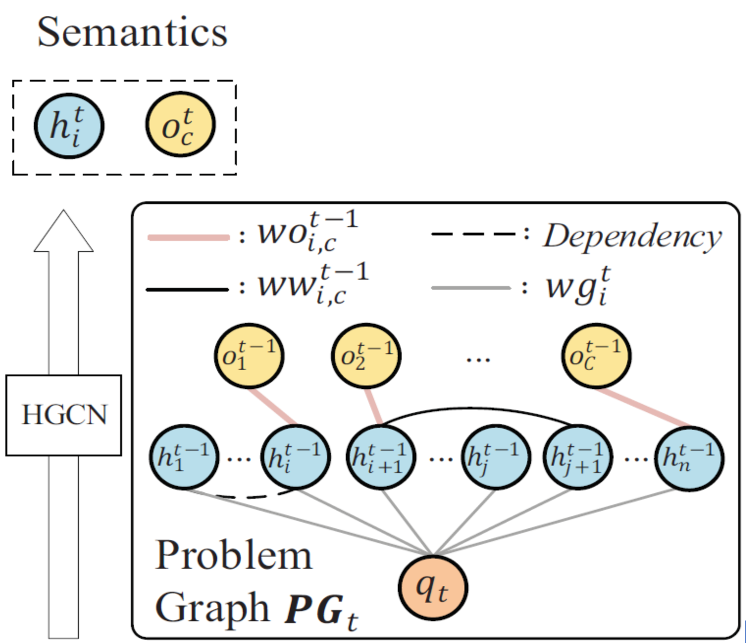


Our Method

Problem: Jack has 3 apples and Amy has 2 bananas, how many fruits do they have?

CogSolver: Knowledge application

- **Decoder**
  - Knowledge-aware module
    - ✓ **Semantics**
      - ✓ HGCN: goal → word → operator



$$AGG_i = [\boxed{wg_i^t \cdot q_t}, \sum_{j=1}^{n} \boxed{ww_{i,j}^{t-1} \cdot h_j^{t-1}}, \sum_{j \in \mathcal{N}_d(i)} \boxed{h_j^{t-1}}],$$

$$h_i' = ReLU(\mathbf{W}_u \cdot AGG_i + b_u),$$

$$f_i = \sigma(\mathbf{W}_{fw} \cdot [h_i^{t-1}, AGG_i] + b_{fw}),$$

$$h_i^t = f_i \cdot h_i^{t-1} + (1 - f_i) \cdot h_i',$$

$$o_c' = ReLU(\mathbf{W}_o \cdot \sum_{i=1}^{n} \boxed{wo_{i,c}^{t-1} \cdot h_i^t} + b_o),$$

$$f_c = \sigma(\mathbf{W}_{fo} \cdot [o_c^{t-1}, \sum_{i=1}^{n} wo_{i,c}^{t-1} \cdot h_i^t] + b_{fo}),$$

$$o_c^t = f_c \cdot o_c^{t-1} + (1 - f_c) \cdot o_c'.$$

Our Method

Problem: Jack has 3 apples and Amy has 2 bananas, how many fruits do they have?

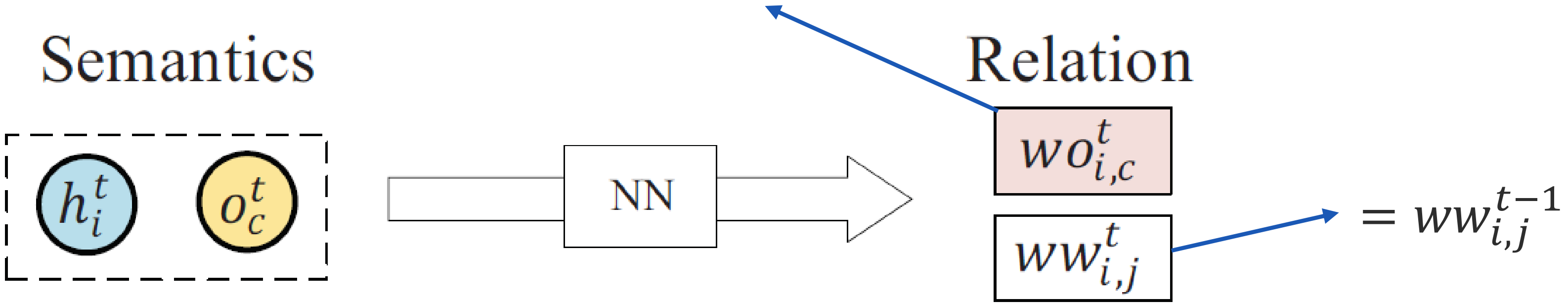
CogSolver: Knowledge application

- **Decoder**
  - Knowledge-aware module
    - ✓ **Relation**

$$wo'_{i,c} = softmax(ReLU(\mathbf{W}_e \cdot [h_i^t, o_c^t, wo_{i,c}^{t-1}] + b_e)),$$

$$f_{i,c} = \sigma(\mathbf{W}_{fe} \cdot [h_i^t, o_c^t] + b_{fe}),$$

$$wo_{i,c}^t = f_{i,c} \cdot wo_{i,c}^{t-1} + (1 - f_{i,c}) \cdot wo'_{i,c}.$$



Our Method

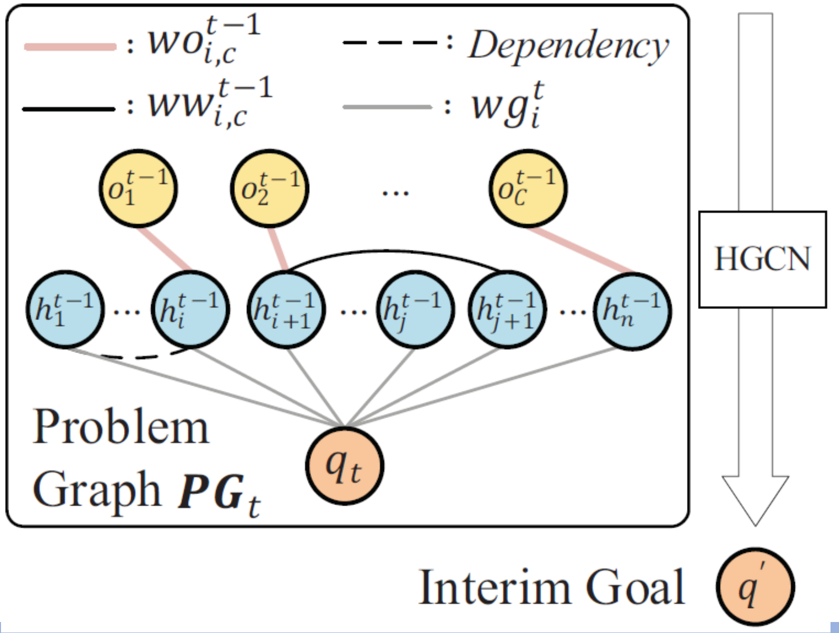
Problem: Jack has 3 apples and Amy has 2 bananas, how many fruits do they have?

CogSolver: Knowledge application

- **Decoder**
  - Knowledge-aware module
    - ✓ **Interim Goal:** q' represents the interim goal of following reasoning
      - ✓ HGCN: Up → Bottom

$$AGG_i = [\sum_{c=1}^{M} wo_{i,c}^t \cdot o_c^t, \sum_{j=1}^{n} ww_{i,j}^t \cdot h_j^t, \sum_{j\in \mathcal{N}_d(i)} h_j^t],$$
$$h_i' = ReLU(\mathbf{W}_u' \cdot AGG_i + b_u'),$$

$$q_a = \sum_{i=1}^{n} wg_i^t \cdot h_i',$$
$$f = \sigma(\mathbf{W}_{fg} \cdot [q_a, q_t] + b_{fg}),$$
$$q' = f \cdot q_t + (1-f) \cdot ReLU(\mathbf{W}_g \cdot q_a + b_g),$$



Our Method

Problem: Jack has 3 apples and Amy has 2 bananas, how many fruits do they have?

CogSolver: Knowledge application

- **Decoder**
  - Symbol prediction
    - ✓ Input: $\{h_i^t\}$, $\{o_c^t\}$, and goal q_t
    - ✓ Output: symbol y_t
    - ✓ Follow pointer-generator network

$$\mathbf{P}(y_t) = \begin{cases} (1-\mathbf{P}_{gen}) \cdot \mathbf{P}_p(y_t) & \text{if } y_t \in N_P, \\ \mathbf{P}_{gen} \cdot \mathbf{P}_g(y_t) & \text{if } y_t \in V_O \cup V_C. \end{cases}$$

Problem: Jack has 3 apples and Amy has 2 bananas, how many fruits do they have?

Our Method

CogSolver: Knowledge application

- **Decoder**
  - Commutative Module
    - ✓ **Motivation:** generate the next goal q_{t+1} while achieving commutative law
    - ✓ If y_t is an operator, q' will be first decomposed to a left sub-goal q^l, and then generate the right sub-goal q^r using q' and left child subtree t^l
    - ✓ Notice: if y_t is $+, \times$, equivalent to generate left goal q^l and q^r
    - ✓ **Core idea**: generate q^l_{inv} by q' and right subtree t^r $\rightarrow$ represents the same meaning with q^l

$$Loss_C = || \underbrace{Decompose2(q', t^r)}_{q^l_{inv}} - \underbrace{Decompose1(q')}_{q^l} ||$$

$$Loss = \sum_{t=1}^{m} - \log \mathbf{P}(y_t \mid y_1, y_2, \cdots, y_{t-1}, P) + \lambda Loss_C$$

Our Method

CogSolver: Knowledge update

- **Semantics**
  - Back-propagation
- **Relation**

$$ww_{i,j} = \sigma(-||w_i - w_j|| + Mean_{dis})$$

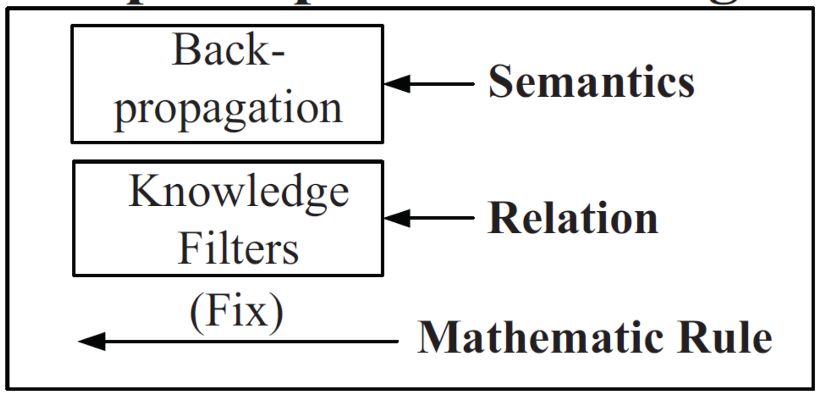
$$wo_{i,c} = softmax(-||w_i - o_c||).$$

$$ww_{i,j} = \begin{cases} ww_{i,j} & \text{if } ww_{i,j} > \delta_1, \\ 0 & \text{else.} \end{cases}$$

$$wo_{i,c} = \begin{cases} wo_{i,c} & \text{if } wo_{i,c} > \delta_2, \\ 0 & \text{else.} \end{cases}$$

- **Mathematical Rule**
  - Fix

Step 3: Update Knowledge



Our Method

Summary

- Before learning, the knowledge in *BRAIN* is initialized randomly
- With the *Store-Apply-Update* steps being iteratively conducted, knowledge is constantly improved and finally the long-term knowledge base is retained
- As the knowledge becomes more and more accurate, the effect of *ARM* to solve problems improves

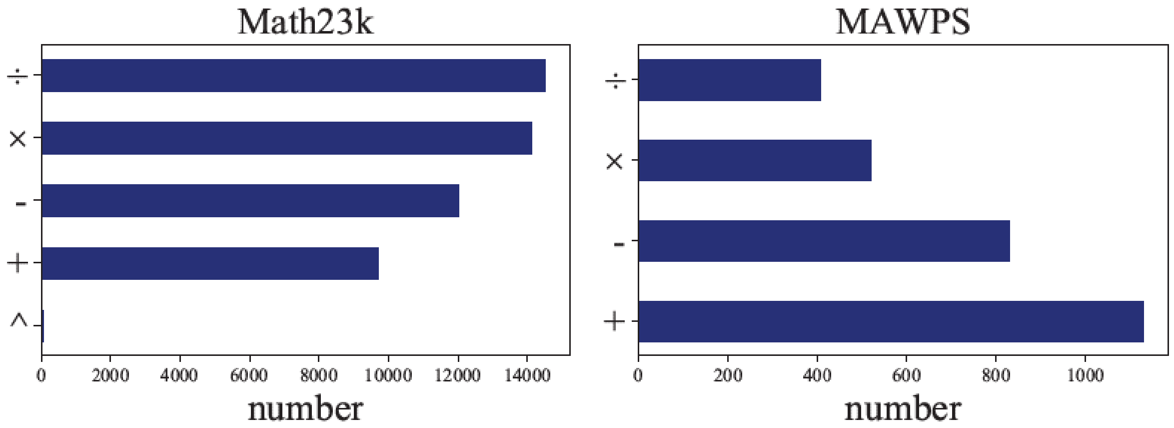
Outline

Experiment

Setups

- Dataset
  - Math23K, MAWPS
- Baseline methods
  - DNS
  - Math-EN
  - T-RNN
  - GROUP-ATT
  - GTS
  - Graph2Tree
  - KA-S2T
  - HMS
- Evaluate metric: Answer Accuracy

Dataset	Math23K	MAWPS
Num. Problems	23,162	2,373
Num. Operators	5	4
Avg. problem length	28.02	30.08



Experiment

Accuracy Performance

	Math23K	MAWPS
DNS	0.581	0.595
Math-EN	0.667	0.692
T-RNN	0.669	0.668
GROUP-ATT	0.695	0.761
GTS	0.743	0.786
KA-S2T	0.763	/[3]
HMS	0.755	0.804
Graph2Tree	0.764	0.820
CogSolver	**0.773***	**0.829***

- CogSolver outperforms all the baselines
- CogSolver can learn knowledge from scratch
- Superior reasoning ability

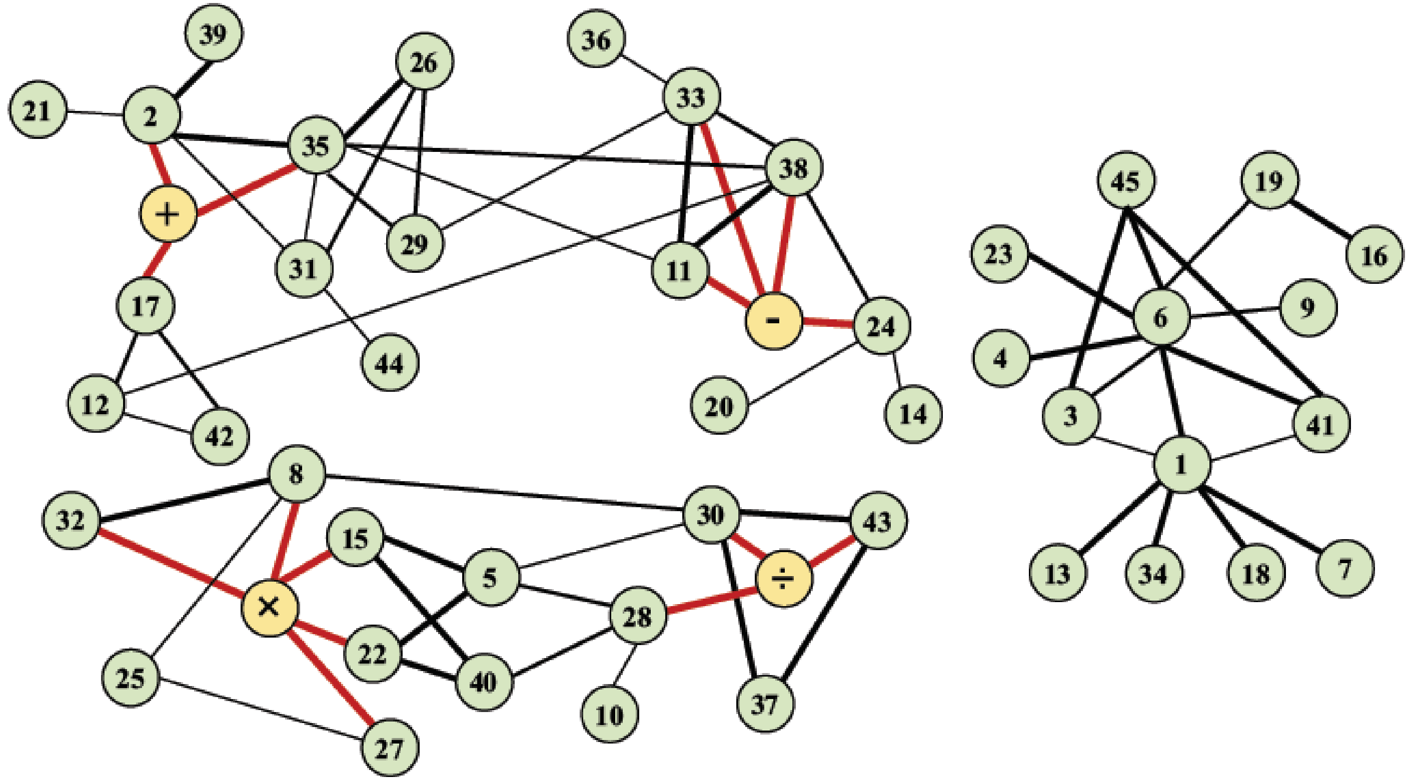
<table>
<tr><th colspan="3">Model</th><th>Math23K</th><th>MAWPS</th></tr>
<tr><td colspan="3">CogSolver</td><td>0.773</td><td>0.829</td></tr>
<tr><td rowspan="2">Store</td><td colspan="2">w/o Relation</td><td>0.754</td><td>0.809</td></tr>
<tr><td colspan="2">w/o Mathematic Rule</td><td>0.766</td><td>0.820</td></tr>
<tr><td rowspan="2">Apply</td><td rowspan="2">w/o Knowledge-aware Module</td><td>Semantics</td><td>0.758</td><td>0.820</td></tr>
<tr><td>Relation</td><td>0.761</td><td>0.818</td></tr>
<tr><td>Update</td><td colspan="2">w/o update Relation</td><td>0.741</td><td>0.796</td></tr>
</table>

- Effectiveness of each component
- Knowledge update is the most important
- Relation knowledge is more effective to reasoning

Experiment

Analysis of BRAIN

- The *BRAIN* after learning on Math23K

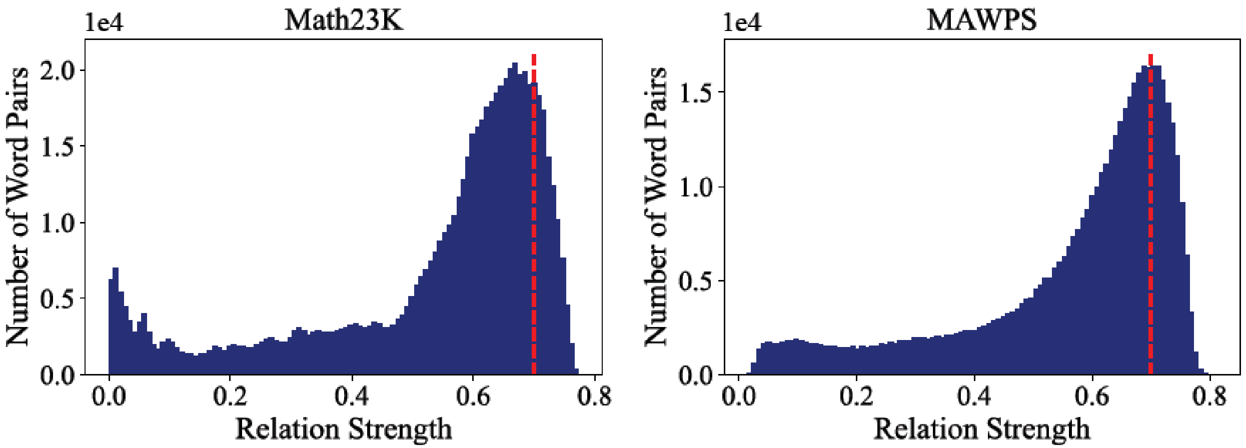


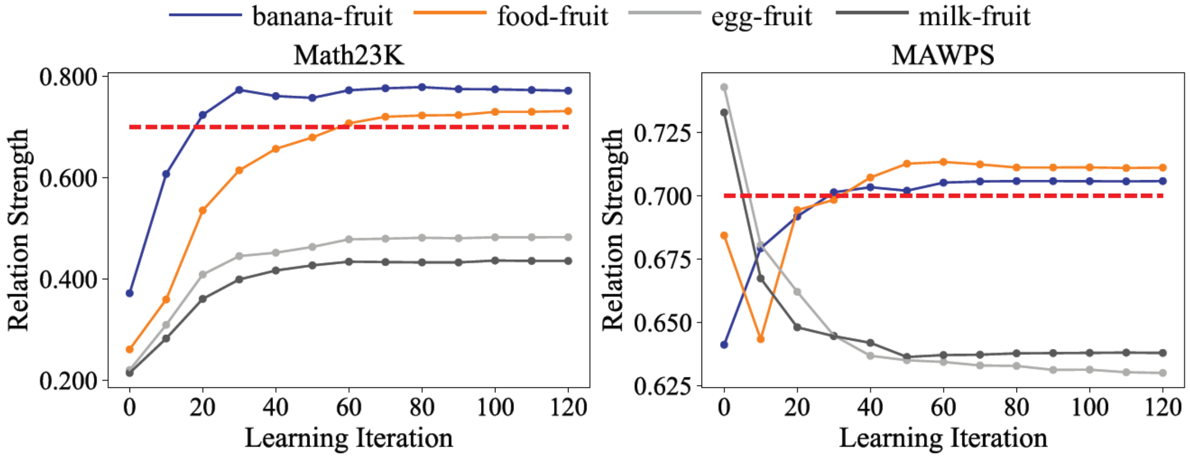
1 fruit	16 clothes	31 earn
2 total	17 remain	32 multiplier
3 pepper	18 pear	33 subtractor
4 dessert	19 goods	34 peach
5 everyone	20 lighten	35 add
6 food	21 bring	36 decimals
7 orange	22 per	37 quotient
8 multiply	23 sesame	38 minus
9 cream	24 lost	39 whole
10 half	25 ratio	40 everyday
11 remainder	26 improve	41 muskmelon
12 others	27 times	42 save
13 banana	28 average	43 divider
14 break	29 exceed	44 include
15 each	30 divide	45 vegetable

Experiment

Analysis of BRAIN

- Word-word relation





- **Observations**
  1. Most of the relationships are distributed around 0.6-0.7
  2. Approximately 77.2% true common-sense knowledge have been obtained by CogSolver

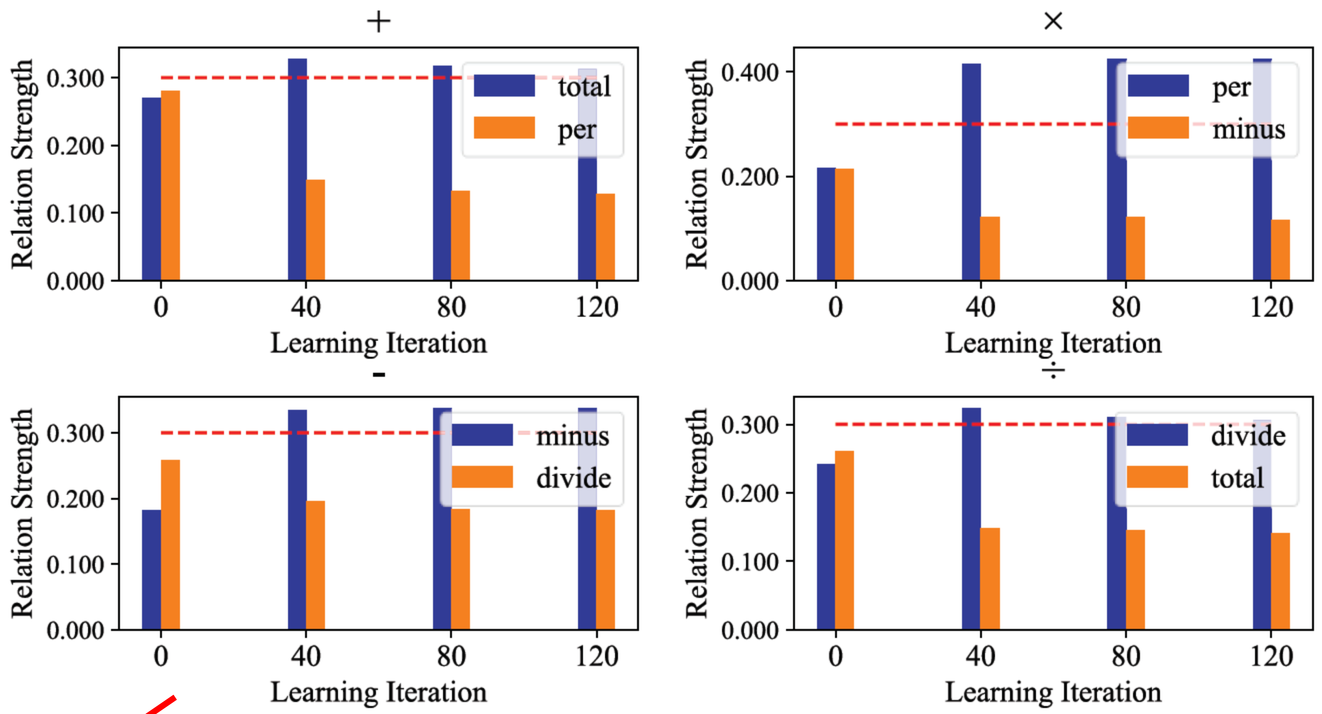
- **Observations**
  1. **Real** relationships gradually strengthen
  2. **fake** relationships are indeed lower than the threshold $\delta_1 = 0.7$

Experiment

Analysis of BRAIN

- Word-operator relation

	Math23k		MAWPS	
	number	top 2	number	top 2
+	3	total,add	569	total,more
−	121	minus,remainder	5	lost,sell
×	16	per,each	60	group,percent
÷	7	divide,average	8	half,split



- CogSolver can learn the reasonable word-operator knowledge correctly

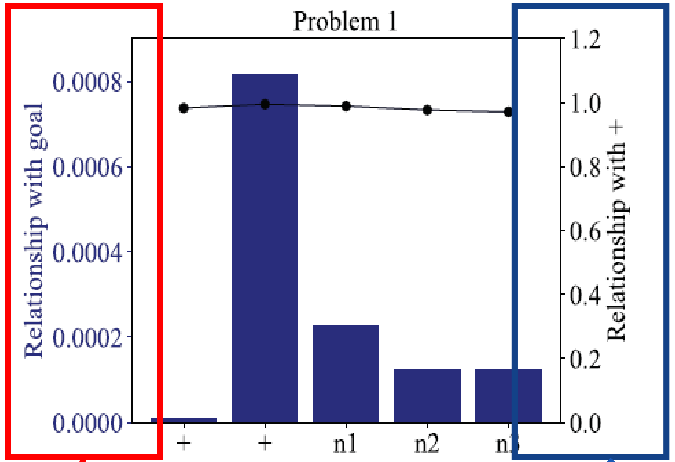
Experiment

Case study

- Aim: illustrate how our CogSolver benefits from learned knowledge to solve a problem

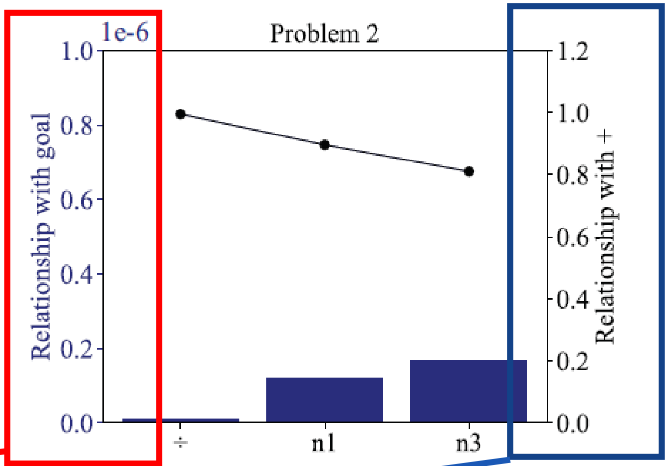
Problem 1: Tim 's cat had kittens . He gave $n1$ to Jessica and $n2$ to Sara . He now has $n3$ kittens . How many kittens did he have to start with ?

Our model: + + $n1$ $n2$ $n3$ (**correct**)
Graph2Tree: + $n1$ $n3$ (**wrong**)



Problem 2: Ruby has $n1$ candies and $n2$ bananas. If she shares the candies among $n3$ friends, how many candies does each friend get ?

Our model: ÷ $n1$ $n3$ (**correct**)
Graph2Tree: − $n1$ $n3$ (**wrong**)



By **paying appropriate attention to "and"** and **relating it with "+"**, CogSolver correctly predicts "+" in problem 1 and 2, while Graph2Tree is unable to utilize such information, thus getting wrong

⟹ **Interpretable reasoning process**

Outline

1 Background

2 Our Method

3 Experiment

4 Conclusion

Conclusion

Summary

- Cognitive Solver (CogSolver) to model knowledge learning process for solving MWP
    - Two systems: *BRAIN-ARM*
    - Three steps: *Store-Apply-Update*
- Experimental results proved the effectiveness and interpretability

Future Work

- Test its performance on other kinds of mathematical problems
- Include pre-trained language models to promote its comprehension ability
- Various knowledge in other fields (e.g., physics)
- …

Thanks for your listening!

For more details, please refer to our paper!

Reporter： Jiayu Liu
jy251198@mail.ustc.edu.cn

中国科学技术大学
University of Science and Technology of China