ELASTIC: Numerical Reasoning with Adaptive Symbolic Compiler

NeurIPS 2022

Jiaxin Zhang, Yashar Moshfeghi

University of Strathclyde

Part 1.

Background

- **Pre-trained Language Models (PLMs)**
  - Show astonishing performance over reading comprehension tasks like SQuAD
  - Fall short of numerical reasoning over text
    - Requires conducting numerical reasoning based on understanding the text

  - Numerical reasoning over text is more challenging than reading comprehension and attracts the interest of the AI community

Zhipeng Xie et al. A goal-driven tree-structured neural model for math word problems. IJCAI. 2019.

Background

- **The numerical reasoning program**
  - An Example requires solving the problem by conducting numerical reasoning

Problem: A small table has a length of 12 inches and a breadth of b inches. Cubes are placed on the surface of the table so as to cover the entire surface. The maximum side of such cubes is found to be 4 inches. Also, a few such tables are arranged to form a square. The minimum length of side possible for such a square is 80 inches. What is the number for b?

(a) Numerical Reasoning Program: $b = \sqrt{(\frac{80}{4})^2 - 12^2}$

(b) Sequential Format:
$\sqrt{((80 \div 4) \times (80 \div 4) - (12 \times 12))}$

(c) Pre-order Traverse Format:
$\sqrt{}, -, \times, \div, 80, 4, \div, 80, 4, \times, 12, 12,$ none

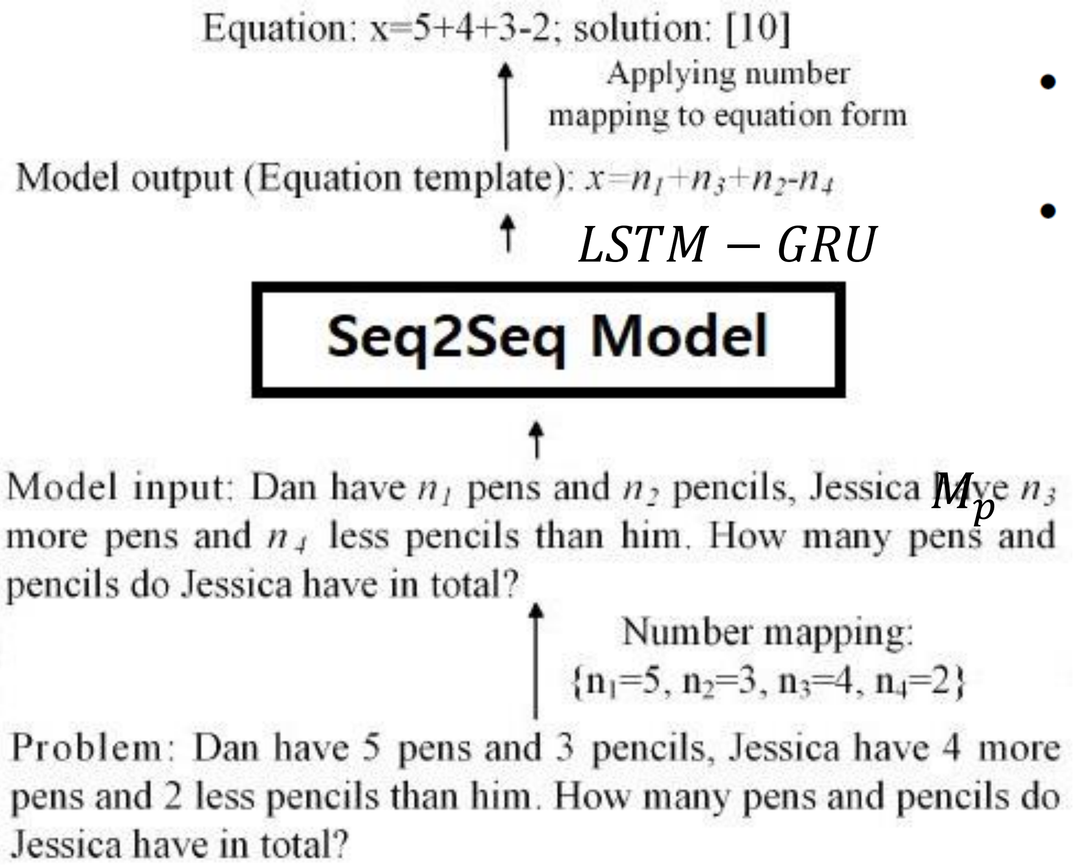
(d) Flattened Format:
divide(80,4)|power(12,const_2)|power(#0,const_2)|subtract(#2,#1)|sqrt(#3)

(e) Nested Format:
sqrt(subtract(power(divide(80, 4), const_2), power(12, const_2)))

Zhipeng Xie et al. A goal-driven tree-structured neural model for math word problems. IJCAI. 2019.

Background

- **Numerical reasoning: The sequential format**
  - Produce invalid expressions such as "3 − ((2)" because of the wrong position of parentheses (Wang et al., 2017)



The five predefined rules

- Rule 1: If r_{t-1} in $\{+, -, *, /\}$, then r_t will not in $\{+, -, *, /,), =\}$;
- Rule 2: If r_{t-1} is a number, then r_t will not be a number and not in $\{(, =\}$;
- Rule 3: If r_{t-1} is "=", then r_t will not in $\{+, -, *, /, =,)\}$;
- Rule 4: If r_{t-1} is "(", then r_t will not in $\{(,), +, -, *, /, =\}$;
- Rule 5: If r_{t-1} is ")", then r_t will not be a number and not in $\{(,)\}$;

Output Probability

$$P(\hat{r}_t|h_t) = \frac{\rho_t \odot e^{h_t^T W^s}}{\sum \rho_t \odot e^{h_t^T W^s}}$$

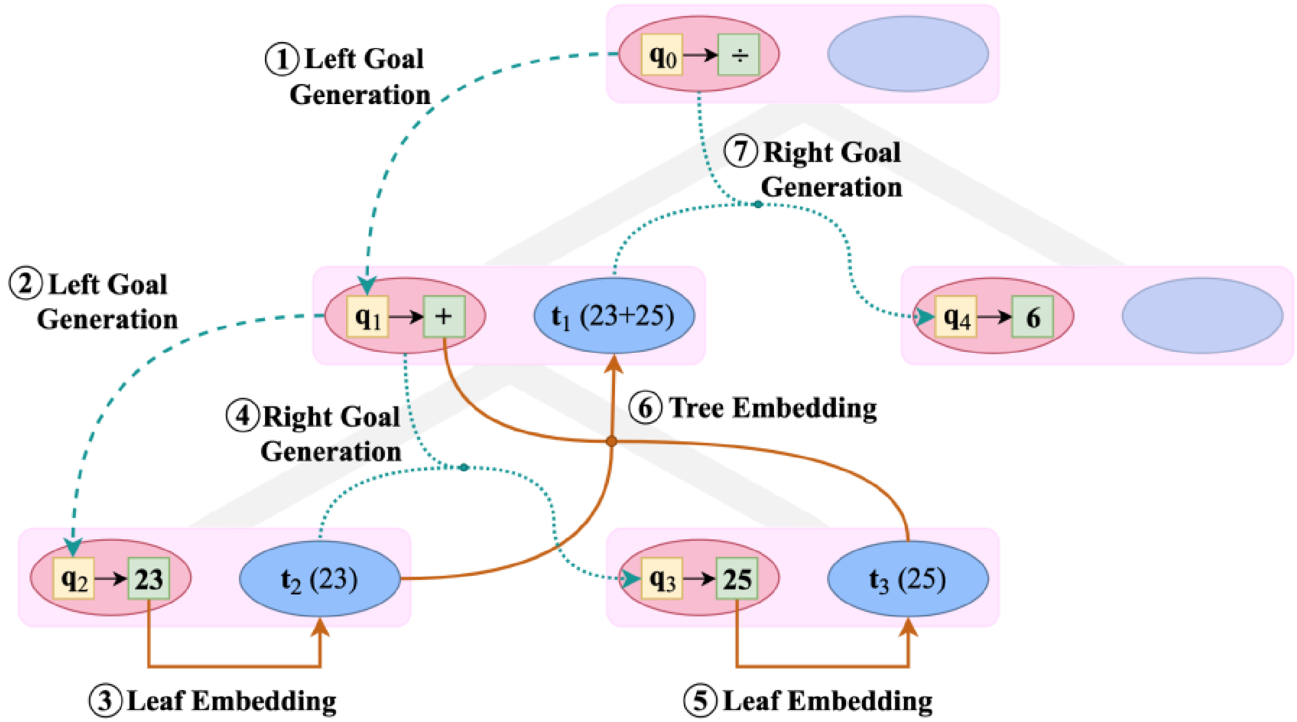
(a) Numerical Reasoning Program: $b = \sqrt{(\frac{80}{4})^2 - 12^2}$

(b) Sequential Format:
$\sqrt{((80 \div 4) \times (80 \div 4) - (12 \times 12))}$

Yan Wang et al. Deep neural solver for math word problems. EMNLP. 2017.

Background

- **Numerical reasoning: The sequence-to-tree architecture**
  - Convert the reasoning program to the binary tree
  - Use the tree-decoder to generate the pre/post-order traversal sequence (Xie et al., 2019)



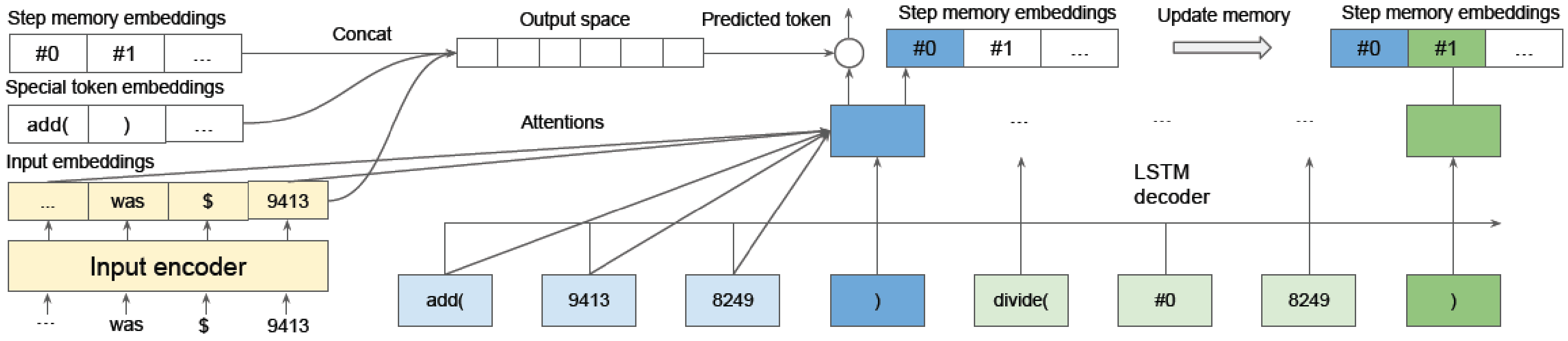
(a) Numerical Reasoning Program: $b = \sqrt{(\frac{80}{4})^2 - 12^2}$

(c) Pre-order Traverse Format:
$\sqrt{}, -, \times, \div, 80, 4, \div, 80, 4, \times, 12, 12,$ none

Zhipeng Xie et al. A goal-driven tree-structured neural model for math word problems. IJCAI. 2019.

Background

- **Numerical reasoning: FinQANet (Chen et al., 2021)**
  - Program Generator: The step memory tokens fmig to denote the results from previous steps, like #0, #1 , etc



Question: **what was the net change in tax positions in 2014?**
Gold program: **add(53818, -36528), add(#0, 157)**

Retrieved evidence:
[1] table row: increases in current period tax positions: 27229 ;
[2] table row: increases in current period tax positions: 53818 ;
[3] table row: balance at december 31 2014: $ 195237 ;
Predicted program:
subtract(27229, 53818)

(a) Numerical Reasoning Program: $b = \sqrt{(\frac{80}{4})^2 - 12^2}$

(d) Flattened Format:
divide(80,4)|power(12,const_2)|power(#0,const_2)|subtract(#2,#1)|sqrt(#3)

Problem Formulation

$$P(A|T,E,Q) = \sum P(G_i|T,E,Q)$$

The general form

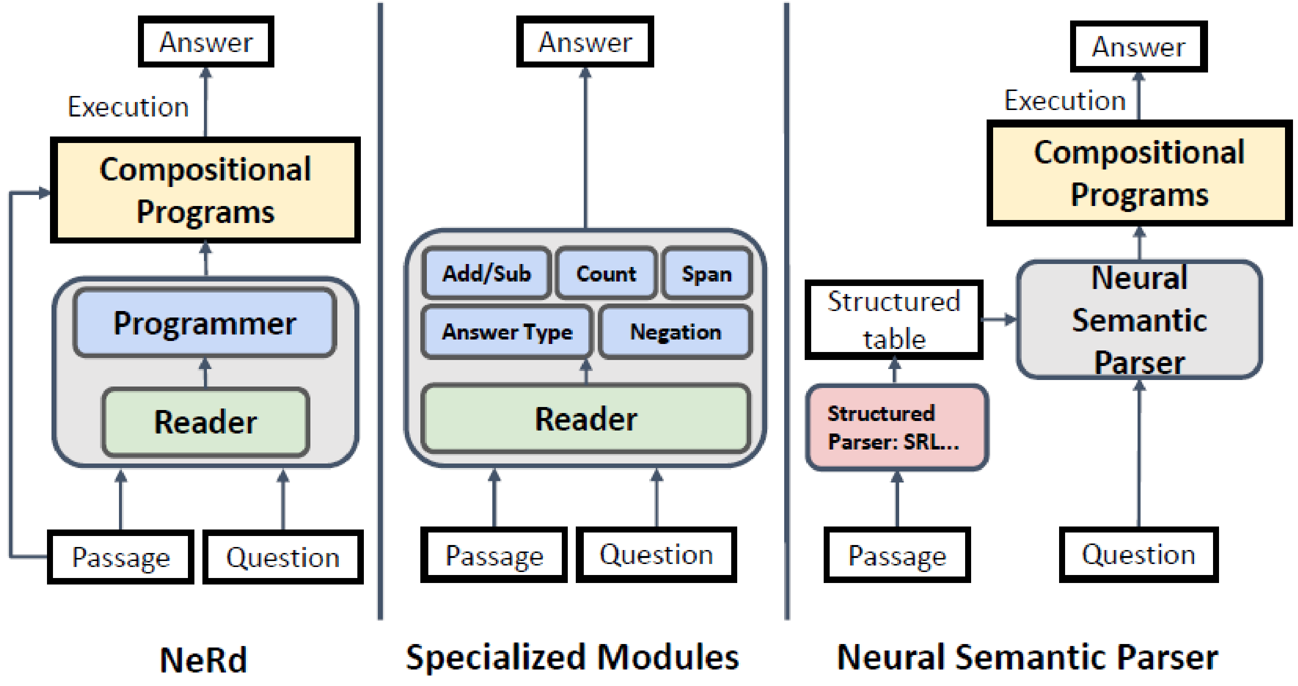
$$\text{op}_1[\text{args}_1], \text{op}_2[\text{args}_2] \ldots, \text{op}_n[\text{args}_n]$$

Zhiyu Chen et al. Finqa: A dataset of numerical reasoning over financial data. EMNLP. 2021.

Part 1.

Background

- **Numerical reasoning: NeRd (Chen et al., 2020)**



Algorithm 1 Hard EM with Thresholding

Input: question-answer pairs $\{(x_i, y_i)\}_{i=1}^N$,
a model p_θ, initial threshold α_0, decay factor γ
for each (x_i, y_i) **do**
  $Z_i \leftarrow \text{DataAugmentation}(x_i, y_i)$
$T \leftarrow 0$
repeat
  $\alpha \leftarrow \alpha_0 * \gamma^T$
  $\mathcal{D} \leftarrow \emptyset$
  for each (x_i, y_i) **do**
    $z_i^* = \arg\max_k p_\theta(z_i^k|x_i), z_i^k \in Z_i$
    if $p_\theta(z_i^*) > \alpha$ or $T = 0$ and $|Z_i| = 1$ **then**
      $\mathcal{D} \leftarrow \mathcal{D} \cup (x_i, z_i^*)$
  Update θ by maximizing $\sum_{\mathcal{D}} \log p_\theta(z^*|x)$
  $T \leftarrow T + 1$
until converge or early stop

(a) Numerical Reasoning Program: $b = \sqrt{(\frac{80}{4})^2 - 12^2}$

(e) Nested Format:
sqrt(subtract(power(divide(80, 4), const_2), power(12, const_2)))

Xinyun Chen et al. Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension. ICLR. 2020.

Background

- **Complicated numerical reasoning problems**
  - Usually contain a long reasoning program, in which the types of operators are diverse, and the number of operands is dynamic
  - Performance is hindered by cascading errors when encountering complicated tasks
    - Most works do not separate the generation of operators and operands

- **Lack extensibility for the operators**
  - Arise from either the flaw of the model architecture or the representation format of the program
  - Make them hard to apply to different data domains

Introduction

- **The numEricaL reASoning with adapTive symbolIc Compiler (ELASTIC)**
  - Separate the generation of operators and operands
    - Allow it to be less influenced by the cascading error from the complicated reasoning

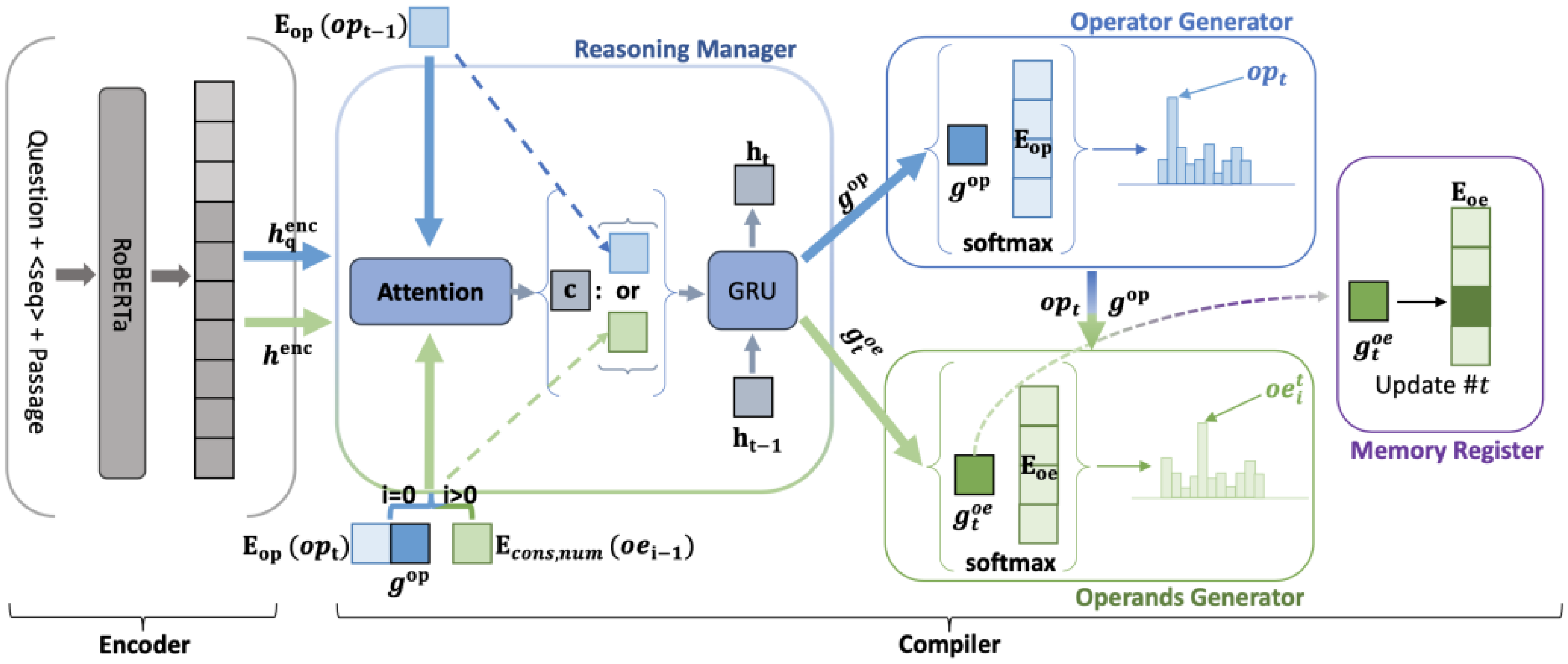
    ➡ Study: Investigate how the length of the numerical reasoning program influences the model's numerical reasoning ability
  - Be adaptable to the number of operands following an operator
    - Make it domain-agnostic to support diverse operators

    ➡ Experiment on FinQA [15], MathQA [19]:
    Different domains - Annual financial reports and GRE/GMAT

Introduction

- **The numEricaL reASoning with adapTive symbolIc Compiler**
  - An Encoder part
    - Extract the contextual representations of the passage and question
  - A Compiler part
    - Generate the numerical reasoning program

ELASTIC

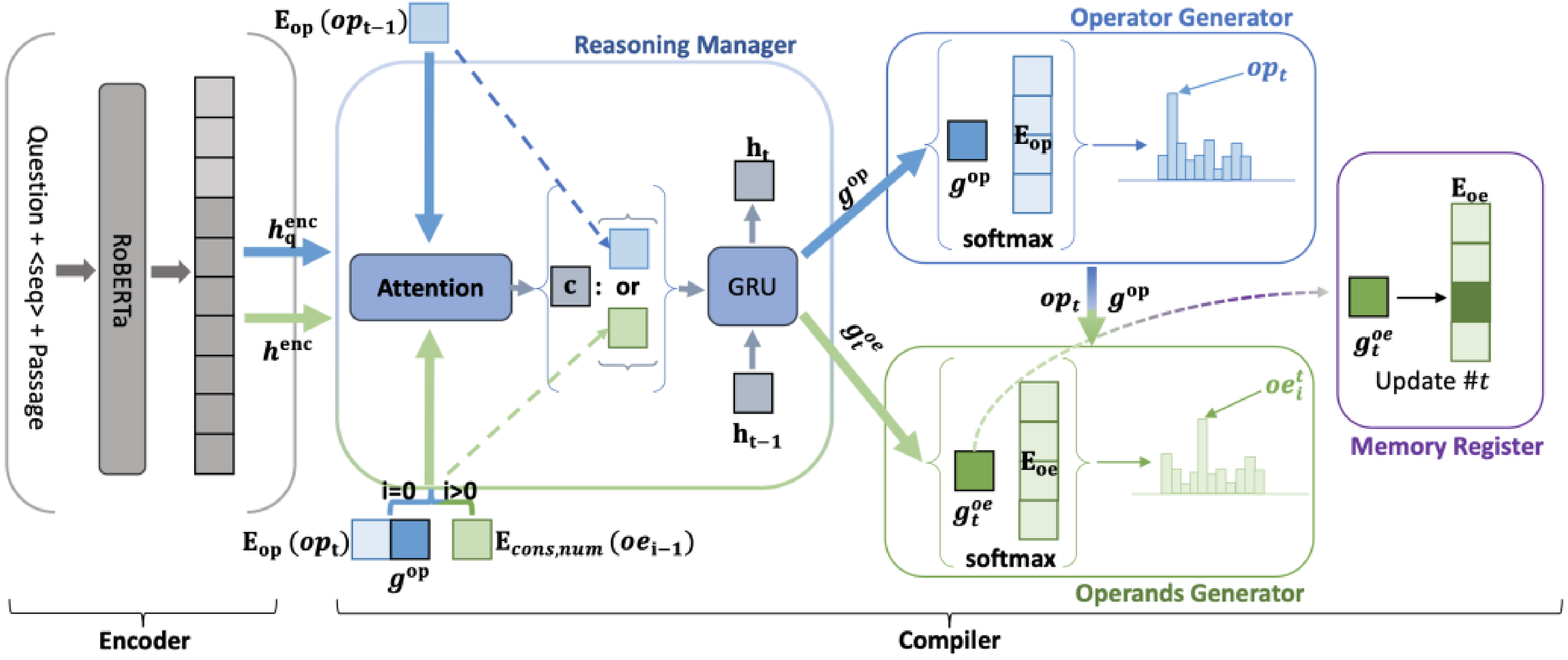


- Maximum Memory Departing Distance (M-MDD)
  - Measure how difficult for the mode to use the executable results from the previous sub-program
  - Use M-MDD to demonstrate the necessity of the Memory Register in ELASTIC

Introduction

- **The numEricaL reASoning with adapTive symbolIc Compiler**
  - An Encoder part
    - Extract the contextual representations of the passage and question
  - A Compiler part
    - Generate the numerical reasoning program

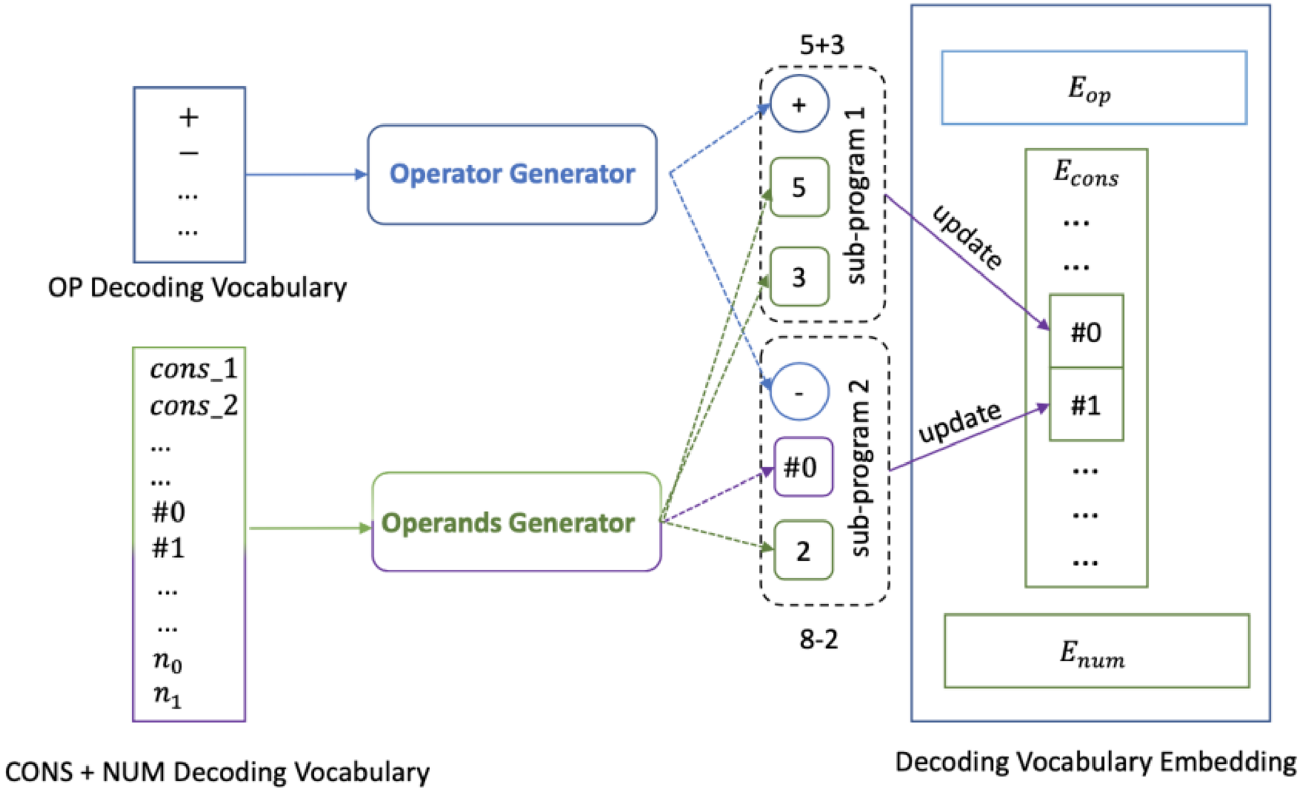
ELASTIC



- Maximum Memory Departing Distance (M-MDD)
  - Measure how difficult for the mode to use the executable results from the previous sub-program
  - Use M-MDD to demonstrate the necessity of the Memory Register in ELASTIC

Introduction

- **The numErical reASoning with adapTive symbolIc Compiler**



- (1) Reasoning Manager sends the guidance vectors g^{op} to the Operator Generator
- (2) Reasoning Manager suspends the Operator Generator, then the Operands Generator takes g_{op}
- (3) When finish the generation of the sub-program r_t, the Memory Register stores the results and updates the embedding vectors of cache token $\#t$ by g_t^{oe}
- (4) Again, the Compiler repeats to generate next sub-program r_{t+1}

Introduction

- **The numEricaL reASoning with adapTive symbolIc Compiler**
  - Present a numerical reasoning model ELASTIC with good adaptability and elasticity
    - Separates the generation of operators and operands
    - Achieve state-of-the-art results on two challenging datasets: FinQA, MathQA
  - Introduce the design of separate modules and Memory Register
    - Make ELASTIC perform stably on complicated numerical reasoning problems
    - The proposed ELASTIC is domain agnostic because it supports diverse operators

Method

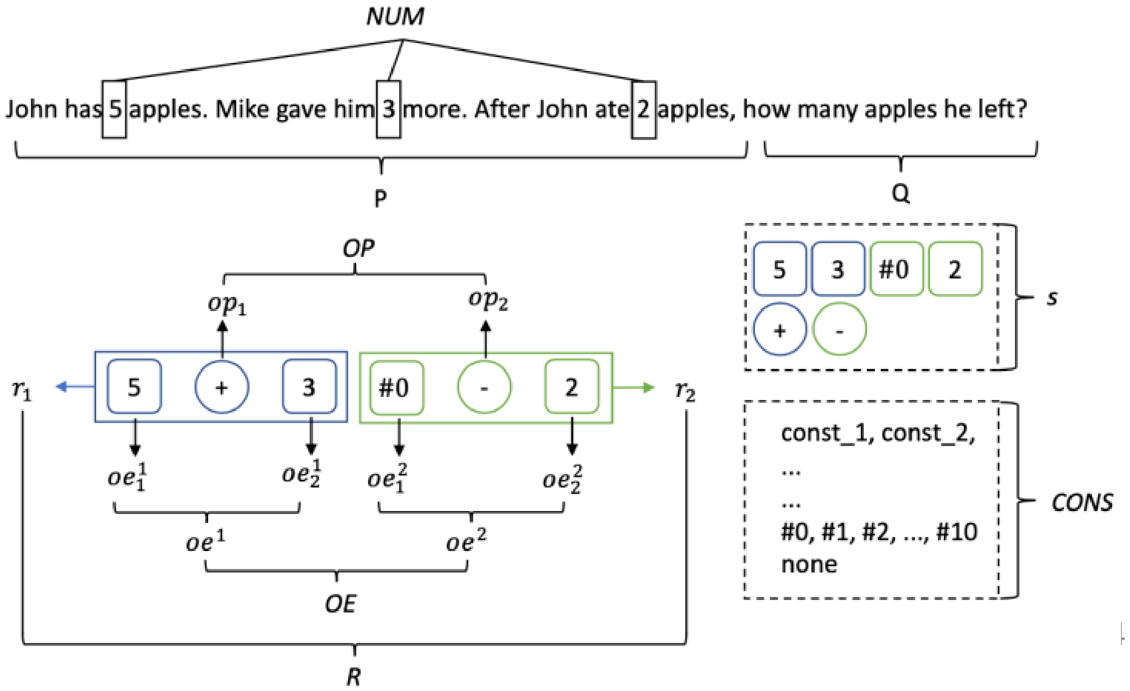
- **Task Definition**
  - The task: generate a numerical reasoning program R
  - R: a sequence of symbols s from operators and operands

The pattern of R

$$R = \left\{ op_i \left[oe^i_j \right]_{j=0}^{m-1} \right\}_{i=0}^{n-1}$$

Notation	Description
P, Q, R	Problem Text, Question Text, Numerical Reasoning Program
NUM	The numbers in P and Q
$CONS$	Constants defined in DSL
OP op_i	All mathematical operators The ith operator in R
OE oe^i oe^i_j	All operands All operands belonging to op_i The jth operands of op_i
s	From either OP or OE, s constitute R
r_i	The i-th sub-program of R $r_i = op_i \left[oe^i \right]$

- Sub-program r
  - A group of one operator and its operands
  - It can be executed since it is a complete program

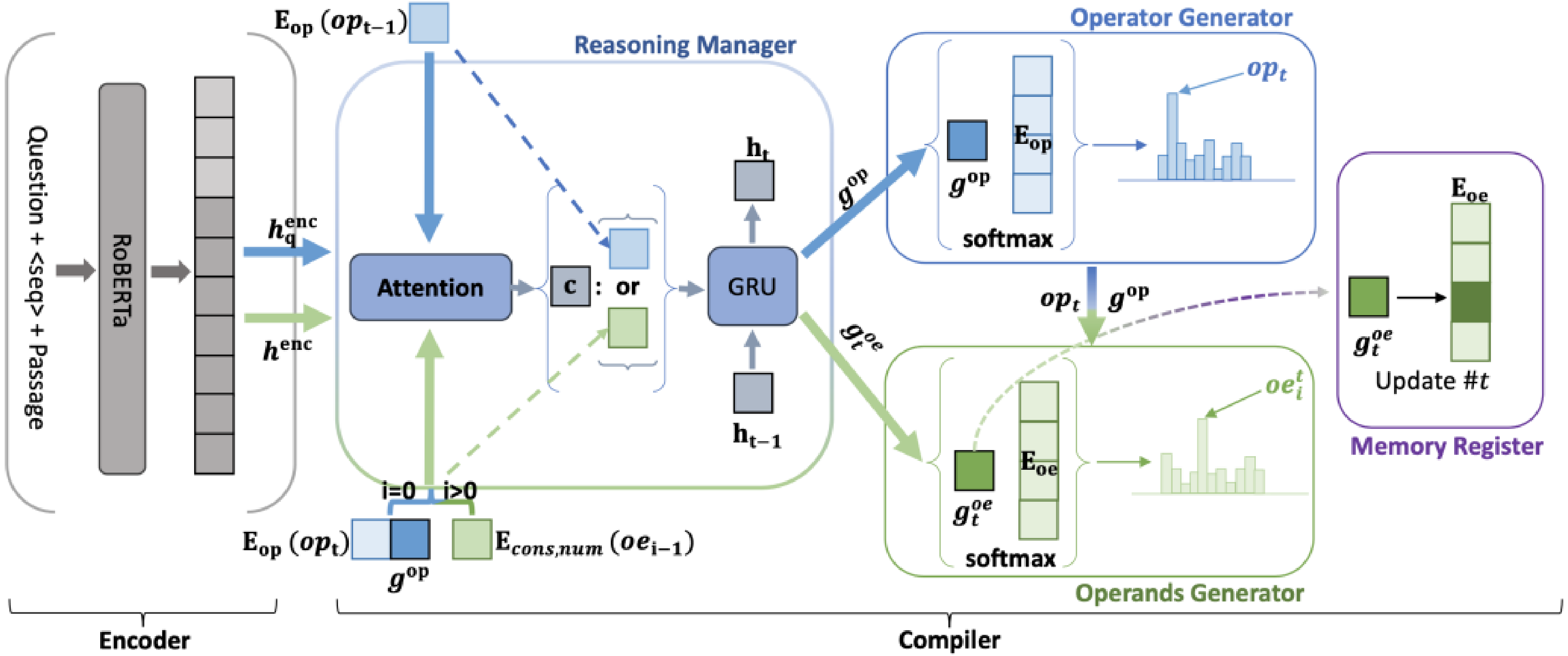


Method

• Encoder Part

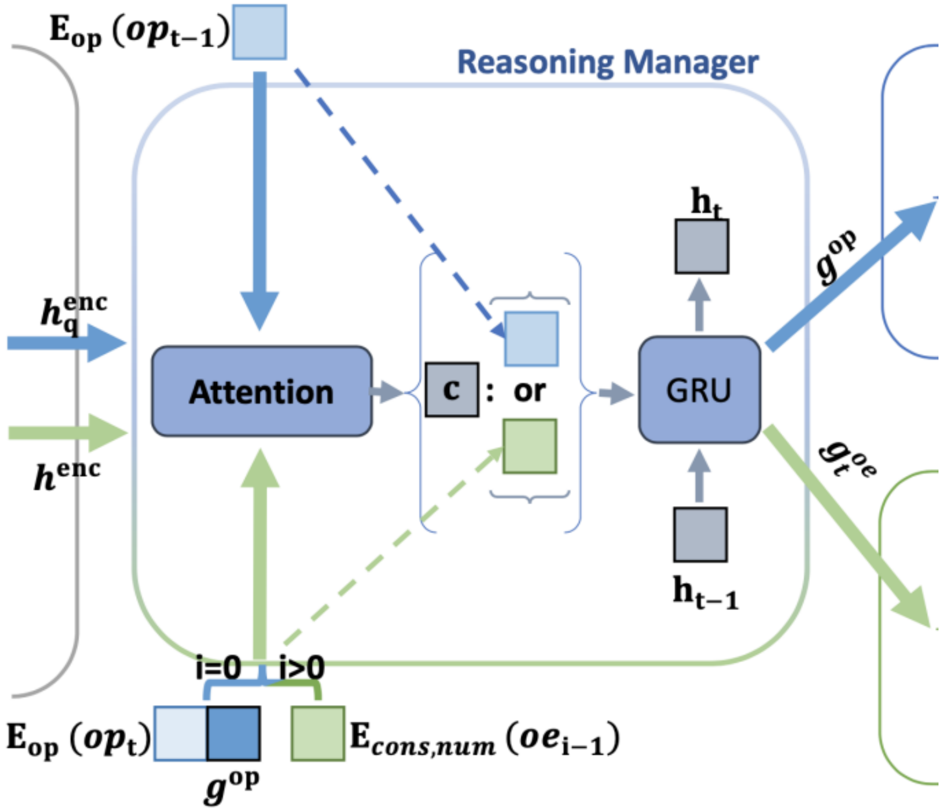
- Take the concatenated sequence of Q and P as input
- Encode the input sequence and outputs the contextual vectors h^{enc}
- The outputs from the final layer of RoBERTa is used as $\mathbf{h}^{enc} \in \mathbb{R}^{h*s}$
- Elastic is not dependent on the specific type of encoder
  - Any model providing contextual vectors of the sequence can be used

ELASTIC



Method

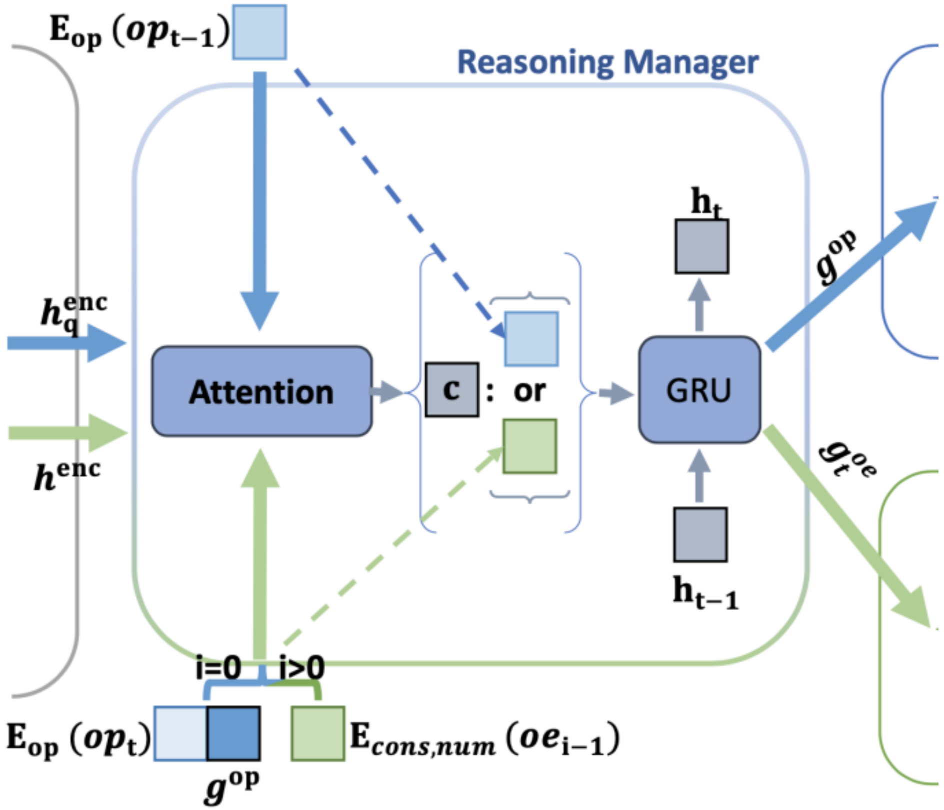
- **Compiler Part: Decoding Vocabulary & Token Embedding**
  - $\mathbf{E}_{op,cons,num}(s)$: The embedding e_s of symbol s of the decoding vocabulary
  - $\mathbf{h}_i^{enc} \in \mathbb{R}^h$: i denotes the index position in the sequence of Q and P



$$\mathbf{e}_s = \begin{cases} \mathbf{E}_{op}(\mathrm{s}) & \text{if } \mathrm{s} \in OP \\ \mathbf{E}_{cons}(\mathrm{s}) & \text{if } \mathrm{s} \in CONS \\ \mathbf{E}_{num}(\mathrm{s}) = \mathbf{h}_i^{enc} & \text{if } \mathrm{s} \in NUM \end{cases}$$

Method

- **Compiler Part: Reasoning Manager**
  - The inputs h^{enc} (Contextual vectors)
    - Calculate the context vector c by the normalized vectors of $\boldsymbol{h_i^{enc}}$ & the attention weights $\boldsymbol{a_i}$
    - c : the encoded information from the Encoder according to the previous generated symbol



$$\mathbf{c} = \sum_i a_i \mathbf{h}_i^{enc}$$

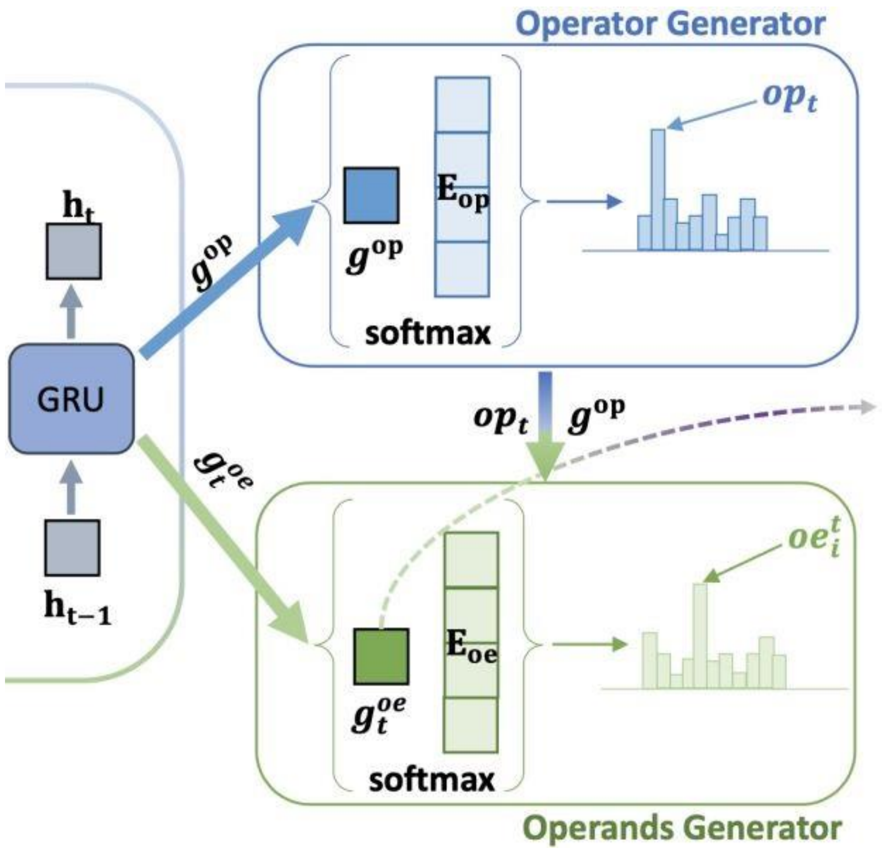
$$a_i = \frac{\exp(\text{score}(\mathbf{e}_{s_{t-1}}, \mathbf{h}_i^{enc}))}{\sum_j \exp(\text{score}(\mathbf{e}_{s_{t-1}}, \mathbf{h}_j^{enc}))}$$

$$\text{score}(\mathbf{e}_{s_{t-1}}, \mathbf{h}_i^{enc}) = \mathbf{e}_{s_{t-1}}^{\mathrm{T}} \mathbf{W}_1 \cdot \mathbf{W}_2 \mathbf{h}_i^{enc} \quad \mathbf{W}_1 \in \mathbb{R}^{h*h} \quad \mathbf{W}_2 \in \mathbb{R}^{h*h}$$

$$\mathbf{g}, \mathbf{h}_t = \text{GRU}(\text{Relu}(\mathbf{W}_3[\mathbf{c} : \mathbf{E}_{op,cons,num}(s_{t-1})]), \mathbf{h}_{t-1})$$

Method

- **Compiler Part: Operator Generator**



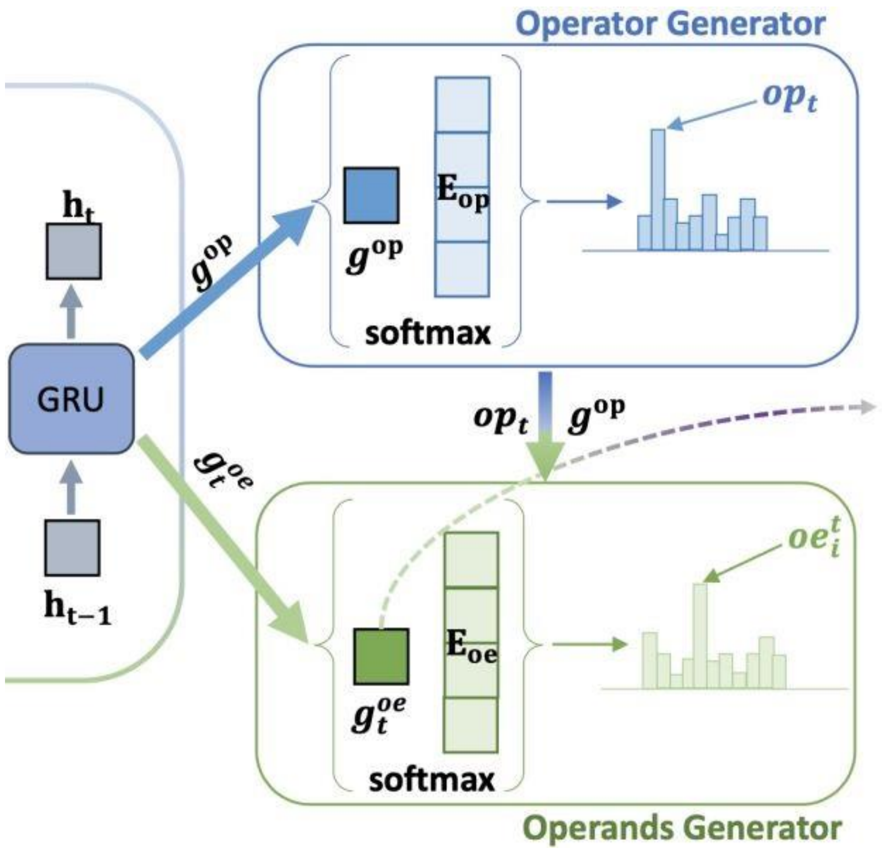
$$\mathbb{P}(i\text{-}op|\mathbf{E}_{op}(op_{t-1}), \mathbf{g}_t^{op}) = \frac{\exp(\mathbf{E}_{op}^{\mathrm{T}}(i\text{-}op)\mathrm{Relu}(\mathbf{W}_{op}\mathbf{g}_t^{op}))}{\sum_{j\text{-}op \in OP} \exp(\mathbf{E}_{op}^{\mathrm{T}}(j\text{-}op)\mathrm{Relu}(\mathbf{W}_{op}\mathbf{g}_t^{op}))}$$

- Input: contextual vectors $\boldsymbol{h_q^{enc}}$
  $\mathbf{E}_{op}(op_{t-1})$ of the previously generated operator

- Reasoning Manager suspends the generation of operators
- ➡ Generates operands $\{oe^t\}$ through the Operands Generator

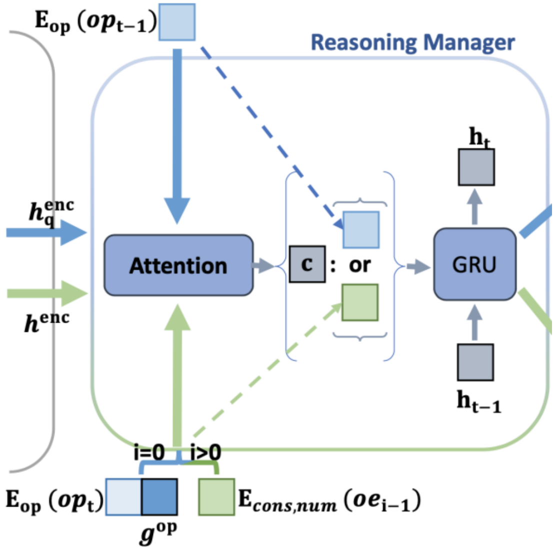
Method

• Compiler Part: Operands Generator



$$\mathbb{P}(i\text{-}oe|\mathbf{E}_{cons,num}(oe_{n-1}^t), \mathbf{g}_t^{oe}) = \frac{\exp(\mathbf{E}_{cons,num}^{\mathrm{T}}(i\text{-}oe)\mathrm{Relu}(\mathbf{W}_{oe}\mathbf{g}_t^{oe}))}{\sum_{j\text{-}oe\in OE}\exp(\mathbf{E}_{cons,num}^{\mathrm{T}}(j\text{-}oe)\mathrm{Relu}(\mathbf{W}_{oe}\mathbf{g}_t^{oe}))}$$

- Input: Different from Operator Generator
  - oe could be a number in Q or P
  - The contextual vectors h^{enc} of all tokens are used
- Initialize the embedding of the initial operand $\mathbf{e}_{(oe_0^t)}$
- Terminate decoding when the token $none$ is produced



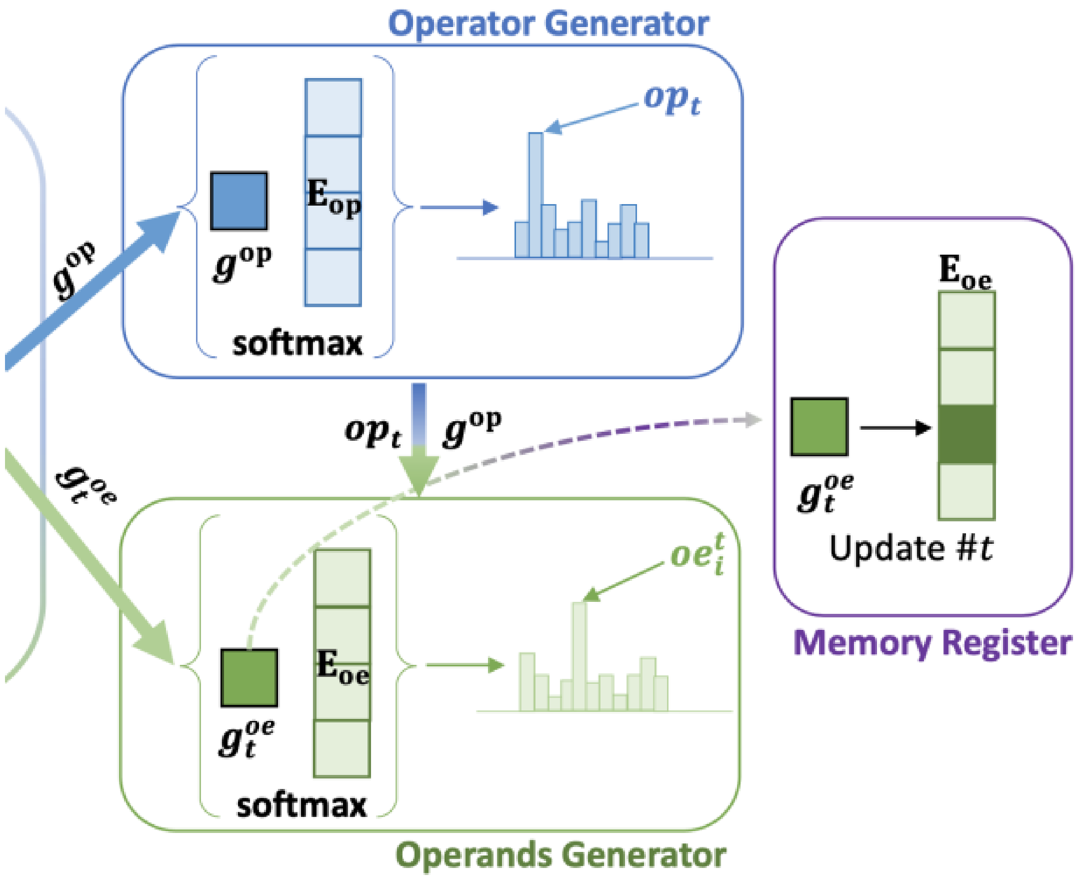
Initial Operand for the Reasoning Manager outputs g_n^{oe}

$$\mathrm{Relu}(\mathbf{W}_4[\mathbf{E}_{op}(op_t) : \mathbf{g}_t]) \; (\mathbf{W}_4 \in \mathbb{R}^{h*2h})$$

The Reasoning Manager outputs g

$$\mathbf{g}, \mathbf{h}_t = \mathrm{GRU}(\mathrm{Relu}(\mathbf{W}_3[\mathbf{c} : \mathbf{E}_{op,cons,num}(s_{t-1})]), \mathbf{h}_{t-1})$$

• Compiler Part: Memory Register



- The operands could be the executable results from the previous sub-program $r_p (p < i)$

- Inspired by FinQA(Chen et al., 2021)

  Introduce a cache token $\#n$ to the $CONS$ of DSL
  - It is used for storing the information of executable results
  - Unlike other constants, $\#n$ does not point to a static value

- It is different according to the different sub-program r_n
  - Update the representation of $\#n$ after the sub-program r_n is generated
  - Update the cache $\#n$ by replacing its embedding with output $\boldsymbol{g}_n^{oe}$
  - **Guidance vector $\boldsymbol{g}_n^{oe}$:** Guide the generation of the last operands of the sub-program r_n

Zhiyu Chen et al. Finqa: A dataset of numerical reasoning over financial data. EMNLP. 2021.

Method

- **Compiler Part: Training Objective**
  - The data $\mathbb{D}$
  - $P_i, Q_i, \hat{op}^i, \hat{oe}^i$

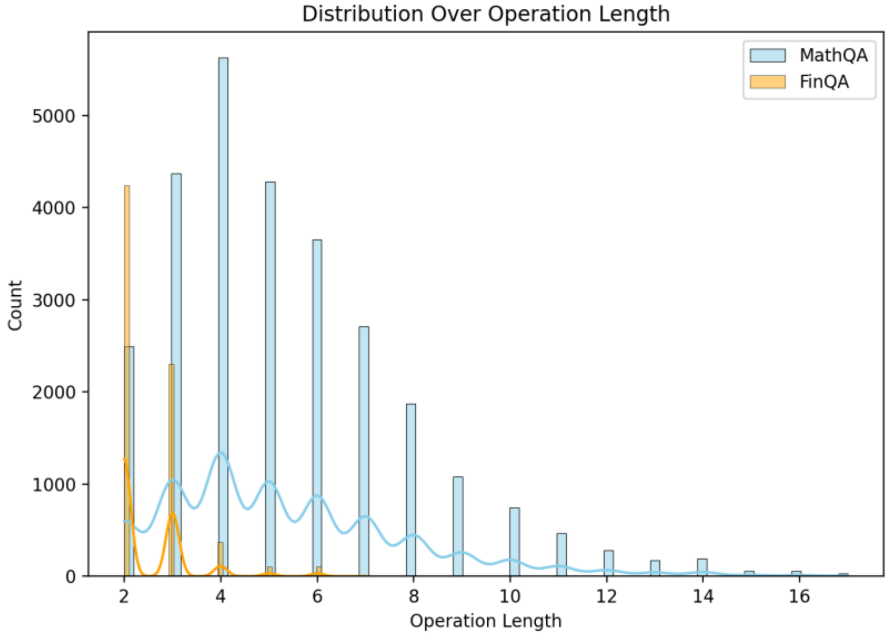
Minimize the sum of the negative log-likelihood

$$\sum_{i=1}^{N} - \{\log \mathbb{P}(\mathrm{OP}^i | \mathrm{P}^i, \mathrm{Q}^i) + \log \mathbb{P}(\mathrm{OE}^i | \mathrm{P}^i, \mathrm{Q}^i)\}$$

Experiments

- **Datasets**
  - FinQA
    - The annual financial reports: 8,281 = 6,251 + 883 + 1,147
    - Execution accuracy (Exe Acc) & Program accuracy (Prog Acc)
    - Only contain operators with two operands

  ➡ Create questions required to be solved by the operators with more than two operands



  - MathQA
    - GRE/GMAT examinations: 37,200 = 29760 + 4,464 + 2976
    - More than 60% have three or more operations
    - GRE questions in many domains including physics, geometry, probability, etc
    - Program accuracy (Prog Acc)

Zhiyu Chen et al. Finqa: A dataset of numerical reasoning over financial data. EMNLP. 2021.
Aida Amini et al. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. NAACL-HLT. 2019.

Experiments

- **Baselines: Sequence-to-sequence (S2S)**
  - FinQANet (Chen et al., 2021)
    - It adopts the Encoder-Decoder architecture with a cache updating mechanism to generate the program
  - NeRd (Chen et al., 2020)
    - It uses the BERT and a pointer-generator-based model to generate the symbolic nested program
  - Graph2Tree (Li et al., 2020)
    - Model the dependency information of the text sequence by the GraphSAGE
    - Generate the program in a tree-structured way
  - NumNet (Ran et al., 2019)
    - Model the numeracy information by a GNN network
    - Train the NumNet+, which replaces the Encoder of the NumNet by RoBERTa-large
    - Wihout Program Accuracy: Do not generate compositional reasoning programs

Zhiyu Chen et al. Finqa: A dataset of numerical reasoning over financial data. EMNLP. 2021.
Xinyun Chen et al. Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension. ICLR. 2020.
Shucheng Li et al. Graphto-tree neural networks for learning structured input-output translation with applications to semantic parsing and math word problem. Findings of ACL. 2020.
Qiu Ran et al. Numnet: Machine reading comprehension with numerical reasoning. EMNLP-IJCNLP. 2019.

Experiments

- **Implementation Details**
  - A server: an NVIDIA Tesla A100 GPU of 40G memory
  - Training epochs: 50(FinQA), 100(MathQA)
  - The batch size: 10
  - Use Adam as optimizer to update the parameters of the models

  - The GRU cell in the decoder
    - The hidden size is the same as the RoBERTa
    - The GRU layers number is 4

  - Inference
    - Use greedy decoding to generate the reasoning program

Experiments

- **Overall Resullts**
  - Outperform the NeRd with a large margin
    - It is worth mentioning that NeRd defines external rules for different operators in their model
  - Outperform the NumNet and NumNet+ by a considerable margin.
    - The internal structure of these models limits their scalability in generating reasoning programs, thus struggles to produce reasoning steps in a systematic manner (Lake et al., 2018)

Datasets & Metrics	FinQA (test)		MathQA (test)
	Exe Acc	Prog Acc	Prog Acc
Graph2Tree	0.37	0.0	69.96†
NumNet	2.32	n/a⋆	n/a⋆
NumNet+	10.29	n/a⋆	n/a⋆
NeRd	52.48‡	49.90‡	79.70†
FinQANet (RoBERTa-base)	60.10†	58.38†	74.12
FinQANet (RoBERTa-large)	65.05†	63.52†	79.20
ELASTIC (RoBERTa-base)	62.66	59.28	82.27
ELASTIC (RoBERTa-large)	**68.96**	**65.21**	**83.00**
Human Expert	91.16†	87.49†	n/a
Human Non-Expert	50.68†	48.17†	n/a

Brenden M. Lake et al. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. ICML. 2018.

Experiments

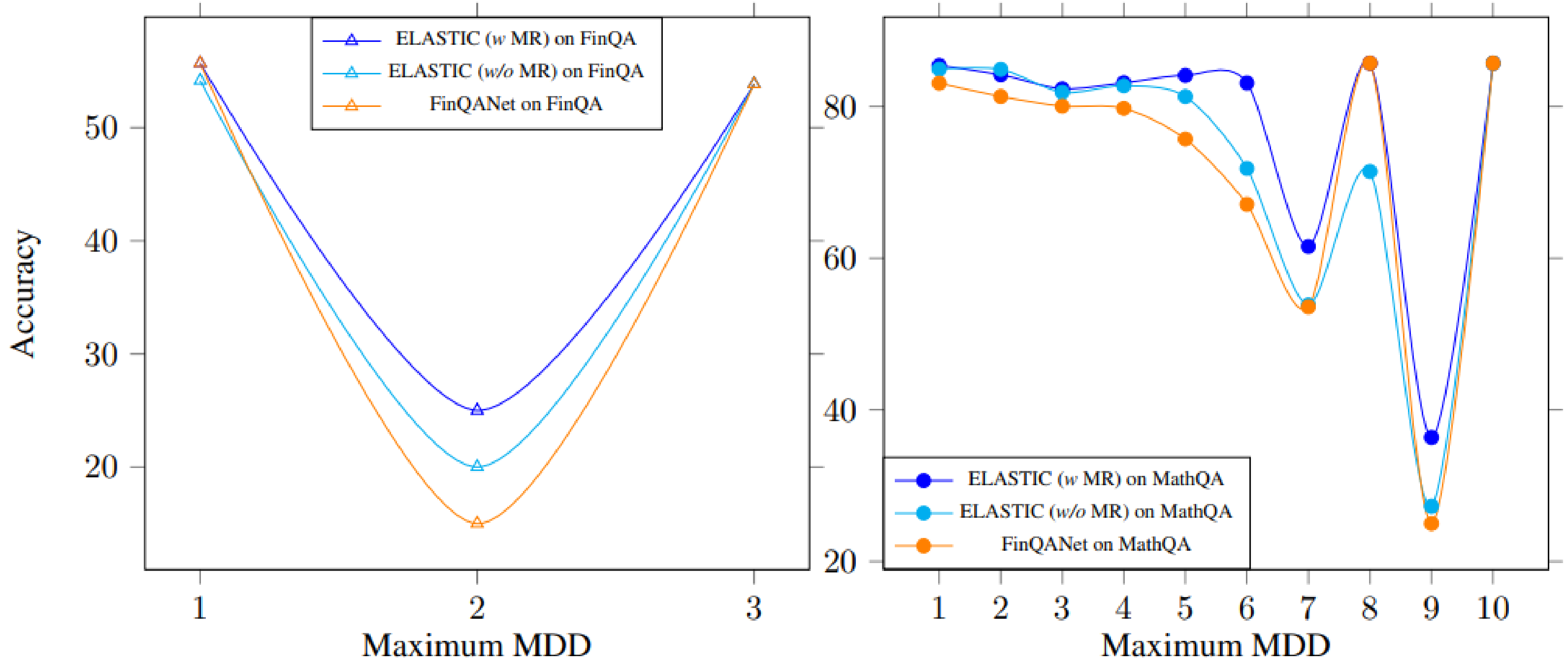
- **Necessity of Memory Register**
  - ELASTIC and FinQANet store the executable results from the previous sub-program in a different way
  - The ELASTIC with Memory Register achieves significantly higher scores than FinQANet on both datasets

Datasets & Metrics	FinQA (test)		MathQA (test)
	Exe Acc	Prog Acc	Prog Acc
ELASTIC *w* MR	**68.96**	**65.21**	**83.00**
ELASTIC *w/o* MR	68.79	64.78	82.68
FinQANet	65.06	63.52	79.20

Experiments

- **Maximum Memory Departing Distance (M-MDD)**
  - The bigger M-MDD is, the more challenging to select the correct previous sub-program result

Program Accuracy

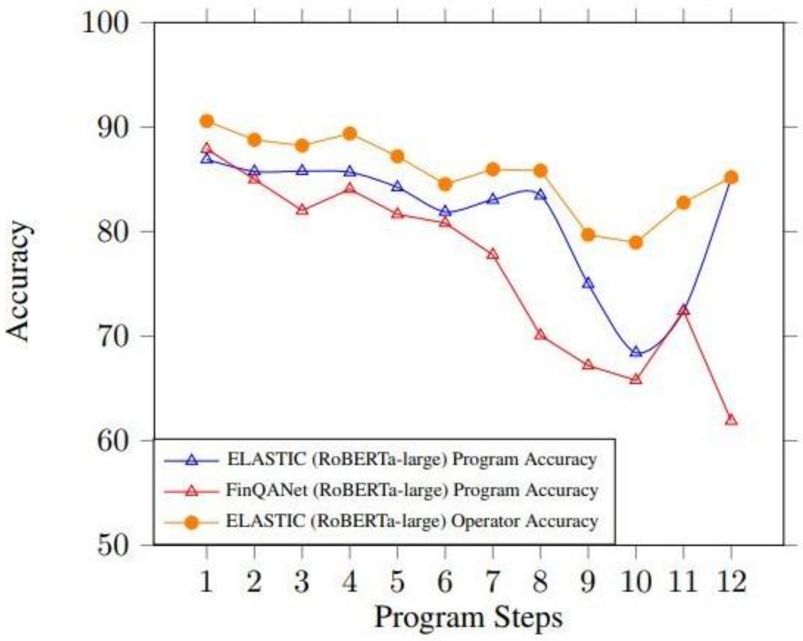


(a) **FinQA** (b) **MathQA**

Experiments

- **Different Program Steps**
  - ELASTIC is less influenced by the cascading error
  - ELASTIC (RoBERTa-large), † original FinQA, ‡ The FinQANet (RoBERTa-large)
  - Program step ≥ 3: FinQA dataset lacks sufficient training examples

Performances on MathQA



(c)

Performances on FinQA

Program Steps	ELASTIC		FinQANet†		FinQANet‡		# Train & Dev
	Exe Acc	Prog Acc	Exe Acc	Prog Acc	Exe Acc	Prog Acc	
=1	76.30	75.66	70.27	68.77	73.70	71.25	4240
=2	66.01	66.01	63.69	61.79	62.34	59.65	2300
≥3	31.78	31.10	31.65	31.65	28.57	23.80	594

Experiments

- **The combination of the original FinQA and extended FinQA**
  - Only contain operators with two operands
    - Create questions required to be solved by the operators with more than two operands
  - ELASTIC (RoBERTa-large) achieves slightly lower scores
  - The state-of-the-art model FinQANet cannot solve the extended FinQA dataset because it can only generate operators with two operands

Results on FinQA

Dataset (Test)	Exec Acc	Prog Acc
original FinQA + extended FinQA	64.5	63.8
extended FinQA	90.0	90.0

Results on Orignal FinQA

Datasets & Metrics	FinQA (test)	
	Exe Acc	Prog Acc
ELASTIC (RoBERTa-base)	62.66	59.28
ELASTIC (RoBERTa-large)	**68.96**	**65.21**

Conclusion

- **The numEricaL reASoning with adapTive symbolIc Compiler (ELASTIC)**
  - Separates the generation of operators and operands, allowing the model to generate the long and complicated reasoning program
  - ELASTIC is domain agnostic and supports diverse operators, increasing adaptability
  - The Memory Register improve the performance of the model by using executable results from the preceding sub-programs

- **Future Work**
  - Improve the accuracy of matching numbers and entities of the text
  - Require annotated reasoning programs, which is labor intensive
  - It is worth investigating how to generate reasoning programs from the trained model

Appendix

- **An example from the extended FinQA dataset**
  - Prediction: The generated numerical reasoning program from ELASTIC (RoBERTa-large).
  - Correctly select the relevant numbers and applies the biggest operator to them
    - 9 numbers (4819, 1703, 1371, 1035, 710, 1903, 1571, 1235, 710) relevant to the "payments“
    - Only selects part of these numbers which are relevant to the "obligations of payments"

Question	What is the biggest obligations of payments between 2007 and 2010?
Passage†	Contractual obligations and commercial commitments the following table (in thousands): The operating lease obligations of payments due by fiscal year total is $4819; The operating lease obligations of payments due by fiscal year 2007 is $1703; The operating lease obligations of payments due by fiscal year 2008 is $1371; The operating lease obligations of payments due by fiscal year 2009 is $1035; The operating lease obligations of payments due by fiscal year 2010 is $710; The total obligations of payments due by fiscal year 2007 is $1903; The total obligations of payments due by fiscal year 2008 is $1571; The total obligations of payments due by fiscal year 2009 is $1235; The total obligations of payments due by fiscal year 2010 is $710.
Prediction	*biggest*(1903, 1571, 1235, 710)

Appendix

- **Case Study: Predictions by ELASTIC (RoBERTa-large) on MathQA**
  - Case 1: Generate the correct numerical reasoning program
    - Use constant none as the padding operand for the operators sqrt and floor
  - Case 2: Separates the generation procedures for operators and operands
    - Prevent the potential interactive distraction between operators and operands
    - Make ELASTIC less liable to being influenced by the cascading error

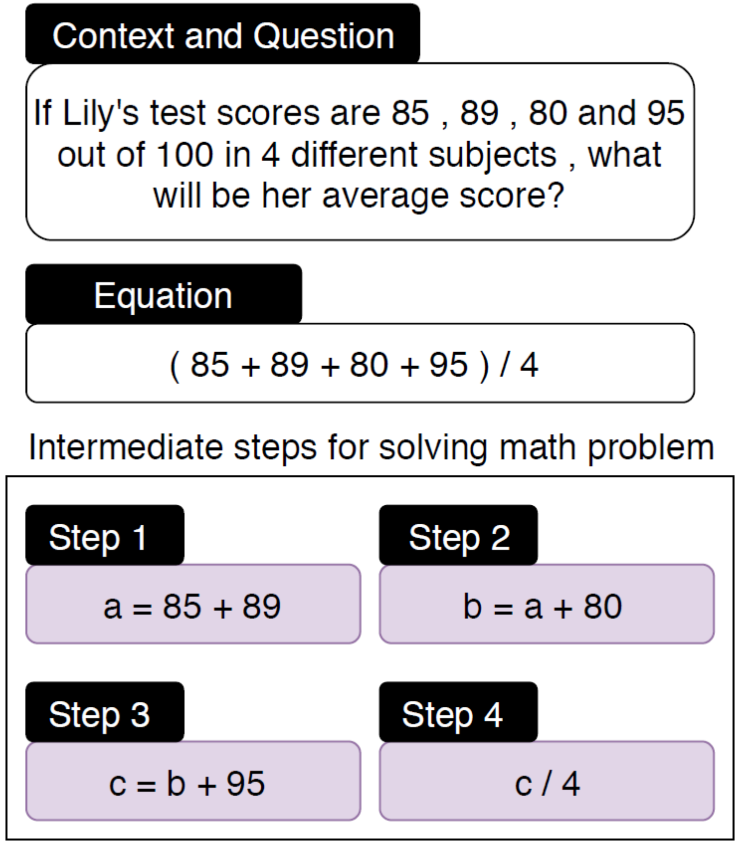
Case 1	Passage	if n is an integer and 101 n ^2 is less than or equal to 10000 , what is the greatest possible value of n ?
	Prediction	*divide*(n2, n0), *sqrt*(#0, none), *floor*(#1, none)
	Golden	*divide*(n2, n0), *sqrt*(#0), *floor*(#1)
Case 2	Passage	Real - Estate salesman z is selling a house at a 25 percent discount from its retail price . Real - Estate salesman x vows to match this price and then offers an additional 5 percent discount. Real - Estate salesman y decides to average the prices of salesmen z and x then offer an additional 30 percent discount. Salesman y's final price is what fraction of salesman x's final price.
	Prediction	*subtract*(const_100,n1), *subtract*(const_100,n0), *subtract*(const_100,n2), *divide*(#0, const_100), *multiply*(#3,#1), *add*(#4,#1), *divide*(#5, const_2), *multiply*(#6,#2), *divide*(#7,const_100), *divide*(#8,#4), *multiply*(#9, const_10)
	Golden	*subtract*(const_100,n1), *subtract*(const_100,n0), *subtract*(const_100,n2), *divide*(#0, const_100), *multiply*(#3,#1), *add*(#4,#1), *divide*(#5, const_2), *multiply*(#6,#2), *divide*(#7,const_100), *divide*(#8,#4), *multiply*(#9, const_10)

Part 6.

Appendix

• MathQA (Amini et al., 2019)

- A new large-scale, diverse dataset of 37k English multiple-choice math word problems
- Require logical reasoning about implied actions and relations between entities



Example of a math word problem

- The complexity of the problem-solving task
  - Deduce implied constants (pi) and knowledge of domain-specific formulas (area of the square)
- Select implied operations and arguments
- Generate a program of intermediate steps for solving a math word problem
- Operations can be dependant to the previous ones by the values they use as arguments

The general form

$$o_1(\mathbf{a}_1)o_2(\mathbf{a}_2)\ldots o_n(\mathbf{a}_n)$$

Operation Sequence generated by model

$$\mathrm{add}_1(85, 89)\mathrm{add}_2(174, 80)$$
$$\mathrm{add}_3(254, 95)\mathrm{divide}_4(349, 4)$$

Aida Amini et al. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. NAACL-HLT. 2019

Part 6.

Appendix

- **FinQANet (Chen et al., 2021)**

Page 91 from the annual reports of GRMN (Garmin Ltd.)

The fair value for these options was estimated at the date of grant using a Black-Scholes option pricing model with the following weighted-average assumptions for 2006, 2005 and 2004:

	2006	2005	2004
Weighted average fair value of options granted	$20.01	$9.48	$7.28
Expected volatility	0.3534	0.3224	0.3577
Distribution yield	1.00%	0.98%	1.30%
Expected life of options in years	6.3	6.3	6.3
Risk-free interest rate	5%	4%	4%

… The total fair value of shares vested during 2006, 2005, and 2004 was $9,413, $8,249, and $6,418 respectively. The aggregate intrinsic values of options outstanding and exercisable at December 30, 2006 were $204.1 million and $100.2 million, respectively. *(… abbreviate 10 sentences ...)*

Question: Considering the weighted average fair value of options , what was the change of shares vested from 2005 to 2006?

Answer: - 400

Calculations:

$$\left(\frac{9413}{20.01} \right) - \left(\frac{8249}{9.48} \right) = -400$$

Program:

divide (9413, 20.01)   divide (8249, 9.48)

substract (#0, #1)

Zhiyu Chen et al. Finqa: A dataset of numerical reasoning over financial data. EMNLP. 2021.