Learning to Reason Deductively: Math Word Problem Solving as Complex Relation Extraction

ACL 2022

Zhanming Jie♥♦, Jierui Li♣♦, Wei Lu♦

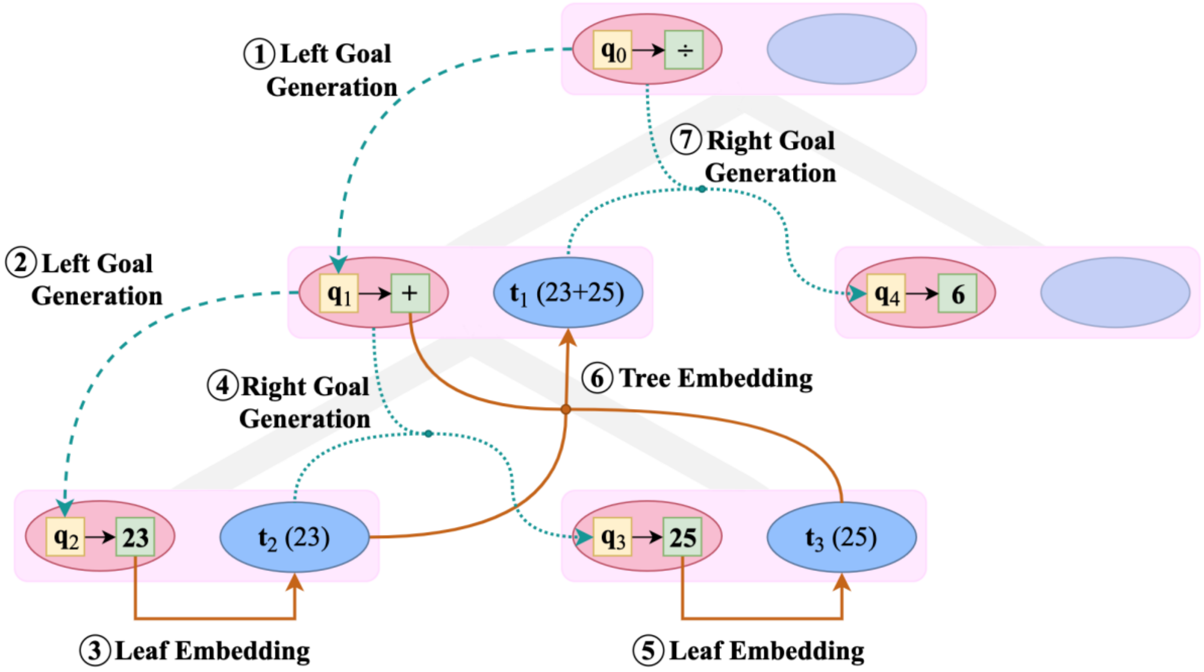
♥ByteDance AI Lab

♣University of Texas at Austin

♦StatNLP Research Group, Singapore University of Technology and Design

Background

- **Math word problem (MWP) solving**
  - A task of answering a mathematical question that is described in natural language
  - Require logical reasoning over the quantities presented in the context
  - Recent research efforts regarded the problem as a generation problem
    - Such model is often represented in the form of a linear sequence or a tree structure (Xie and Sun, 2019)



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

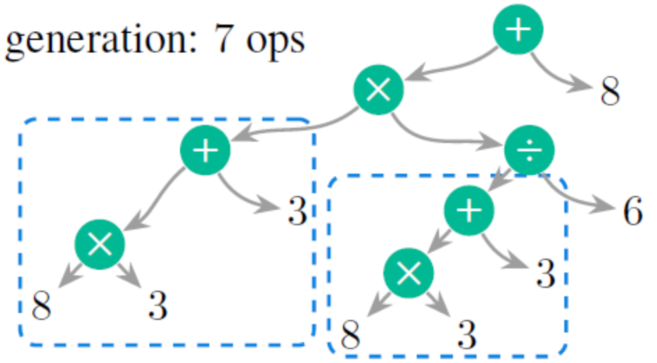
Background

A MWP example taken from MathQA

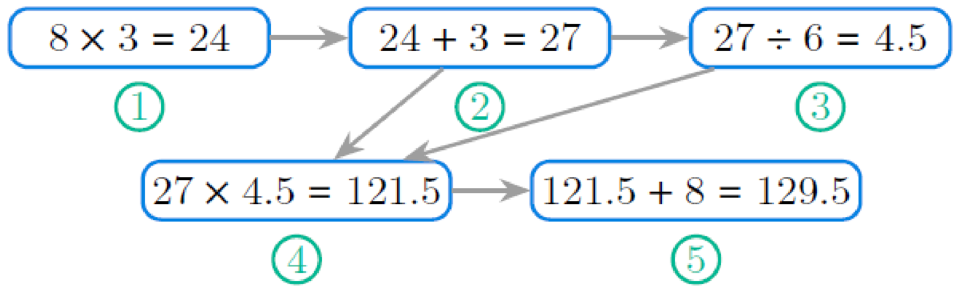
Question: *In a division sum , the remainder is* ***8*** *and the divisor is* ***6*** *times the quotient and is obt- -ained by adding* ***3*** *to the thrice of the remainder. What is the dividend?*

Answer: 129.5 **Expr**: $((8 \times 3 + 3) \times (8 \times 3 + 3) \div 6) + 8$

Tree generation: 7 ops



Our deductive procedure: 5 ops



- **A structure generation approach**
  - Generate the target expression in the form of a tree structure
- **Limitation**
  - Such a process typically involves a particular order when generating the structure
  - Given the complexity of the problem
  - The decision at first step: ("+") operation
    - The decision could be counter-intuitive
    - Does not provide adequate explanations that show the reasoning process when being presented to a human learner
  - Identical sub-trees ("$8 \times 3 + 3$")
    - Require intoducing a certain specifically designed mechanism for reusing the already generated intermediate expression
    - Prevent model repeating the same effort in its process for generating the same sub-expression

Introduction

- **Deductive reasoning**
  - One of the important abilities in children's cognitive development (Piaget, 1952)
    - Investigations on the ability to coordinate corresponding sets and a study of the cardinal and ordinal aspects of numbers and their interrelationships
    - Deal with the child's growing awareness of basic additive and multiplicative properties of numbers

Jean Piaget. 1952. Child's Conception of Number. Routledge. https://www.wiley.com/en-au/The+Child's+Conception+of+Number-p-9780393003246

Part 2.

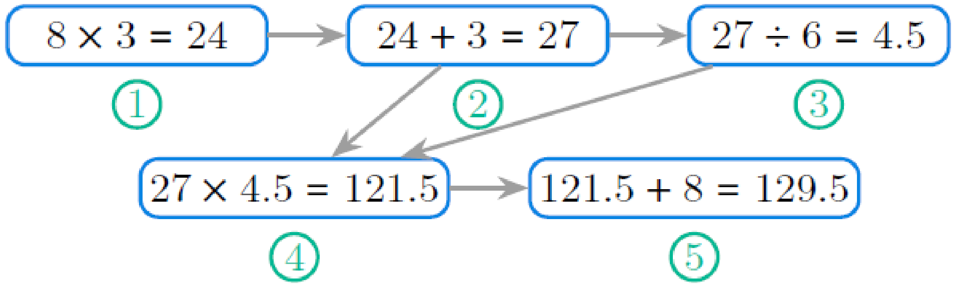
Introduction

- **An approach that explicitly presents deductive reasoning steps**
  - 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
  - Learn how to handle the new quantities that emerge from the intermediate expressions
  - Effectively search for the optimal sequence of operations (relations)

A MWP example taken from MathQA

Question: *In a division sum , the remainder is* ***8*** *and the divisor is* ***6*** *times the quotient and is obt- -ained by adding* ***3*** *to the thrice of the remainder. What is the dividend?*

Our deductive procedure: 5 ops



Relation extraction between two chosen quantities

- Directly extracts the relation ("×") between 8 and 3
- Context: ("remainder is 8", "thrice of the remainder")
- Allows us to reuse the results from the intermediate expression in the fourth step

Introduction

- **Contributions**
  - Formulate MWP solving as a complex relation extraction task
    - Aim to repeatedly identify the basic relations between different quantities
    - The first effort that successfully tackles MWP solving from such a new perspective
  - Automatically produce explainable steps that lead to the final answer
  - Experimental results
    - Our model significantly outperforms existing strong baselines
    - The model performs better on problems with more complex equations than previous approaches

Task Definition

- **Task Definition**
  - Require voking a relation classification module at each step, yielding a deductive reasoning process
    - Given a problem description $\mathcal{S} = \{w_1, w_2, \cdots, w_n\}$ that consists of a list of n words and $\mathcal{Q}_S = \{q_1, q_2, \cdots, q_m\}$
    - List of m quantities that appear in $\mathcal{S}$, our task is to solve the problem and return the numerical answer
    - Each of the primitive mathematical operations ("+", " $-$ ", " $\times$ ", " $\div$ ", " $**$ ") above can essentially be used for describing a specific relation between quantities
  - Some questions cannot be answered without relying on certain predefined constants
    - The constants (such as π and 1) may not have appeared in the given problem description
    - Therefore consider a set of constants $\mathcal{C} = \{c_1, c_2, \cdots, c_{|\mathcal{C}|}\}$
    - Such constants are regarded as quantities (i.e., $\{q_{m+1}, q_{m+2}, \ldots, q_{m+|\mathcal{C}|}\}$)
    - May play useful roles when forming the final answer expression

A Deductive System

- **A Deductive System**
  - Relation (e.g., " + ") between two quantities yields an intermediate expression e
  - At step t, the expression $e^{(t)}$ becomes a newly created candidate quantities
  - One of candidate quantities is ready for deductive reasoning step $t+1$

Initialization

$$\mathcal{Q}^{(0)} = \mathcal{Q}_{\mathcal{S}} \cup \mathcal{C}$$

At step t

$$e_{i,j,op}^{(t)} = q_i \xrightarrow{op} q_j \quad q_i, q_j \in \mathcal{Q}^{(t-1)}$$

$$\mathcal{Q}^{(t)} = \mathcal{Q}^{(t-1)} \cup \{e_{i,j,op}^{(t)}\}$$

$$q_{|\mathcal{Q}^{(t)}|} := e_{i,j,op}^{(t)}$$

Initialization

input: q in $\mathcal{Q}^{(0)}$

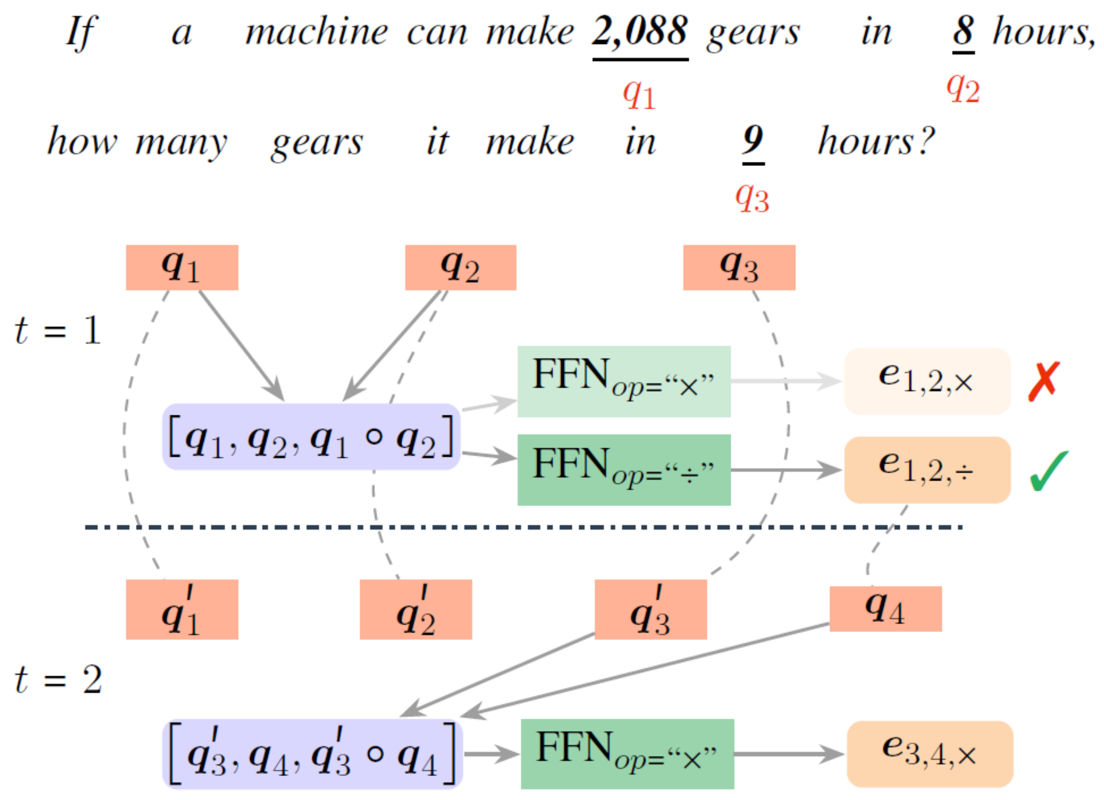
axiom: $0 : \langle q_1, \cdots, q_{|\mathcal{Q}^{(0)}|} \rangle$

$$q_i \xrightarrow{op} q_j : \frac{t : \langle q_1, \cdots, q_{|\mathcal{Q}^{(t-1)}|} \rangle}{t+1 : \langle q_1, \cdots, q_{|\mathcal{Q}^{(t-1)}|} \mid q_{|\mathcal{Q}^{(t)}|} := e_{i,j,op}^{(t)} \rangle}$$

- $e_{i,j,op}^{(t)}$: The expression after applying the relation op to the ordered pair (q_i, q_j)

Part 5.

Model Components

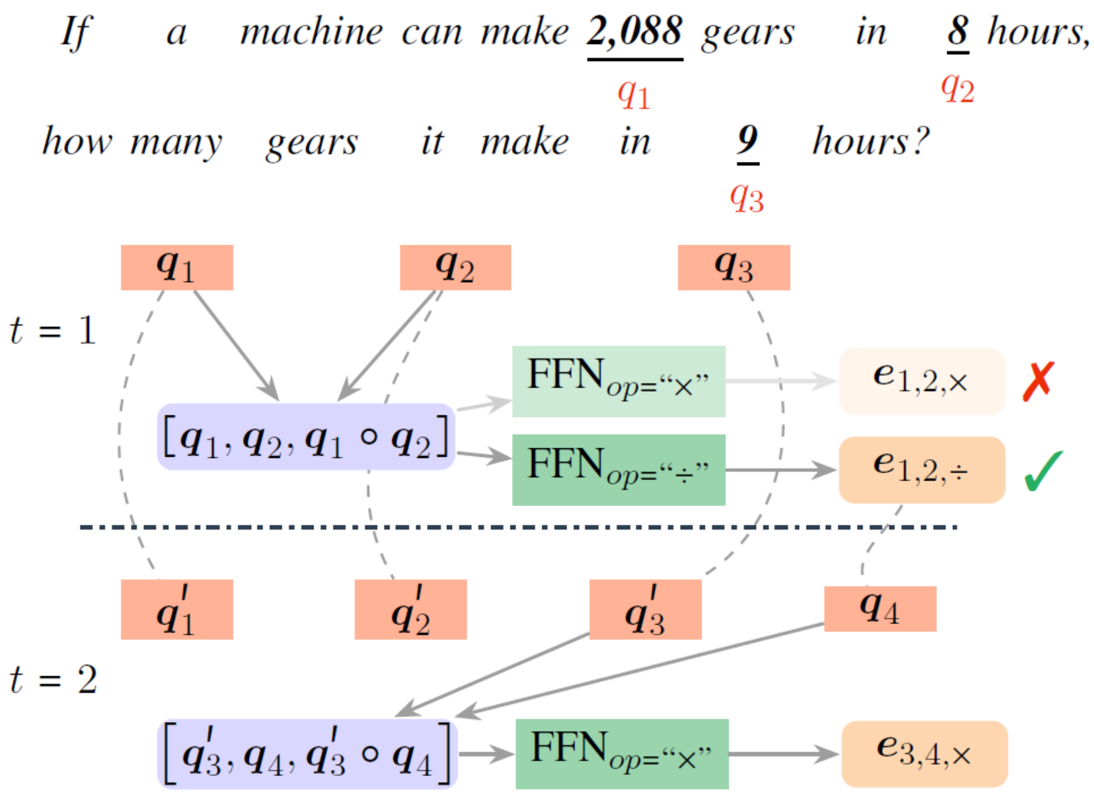


Model architecture for the deductive reasoner

$$“q_1 \div q_2 \times q_3”$$

- **Reasoner**
  - Convert the quantities (e.g., 2,088) into a general quantity token “ $< quant >$ "
  - Adopt a pre-trained language model such as BERT or ROBERTa
    - Obtain the quantity representation q for each quantity q

Model Components



Model architecture for the deductive reasoner

$$\text{“}q_1 \div q_2 \times q_3\text{”}$$

- **Reasoner**
  - Similar to Lee et al. (2017)
  - Obtain the representation of quantity pairs (q_i, q_j)
    - Concatenate the two quantity representations and the element-wise product between them
  - A non-linear feed-forward network (FFN) on top of the pair representation
    - Get representation of newly created expression

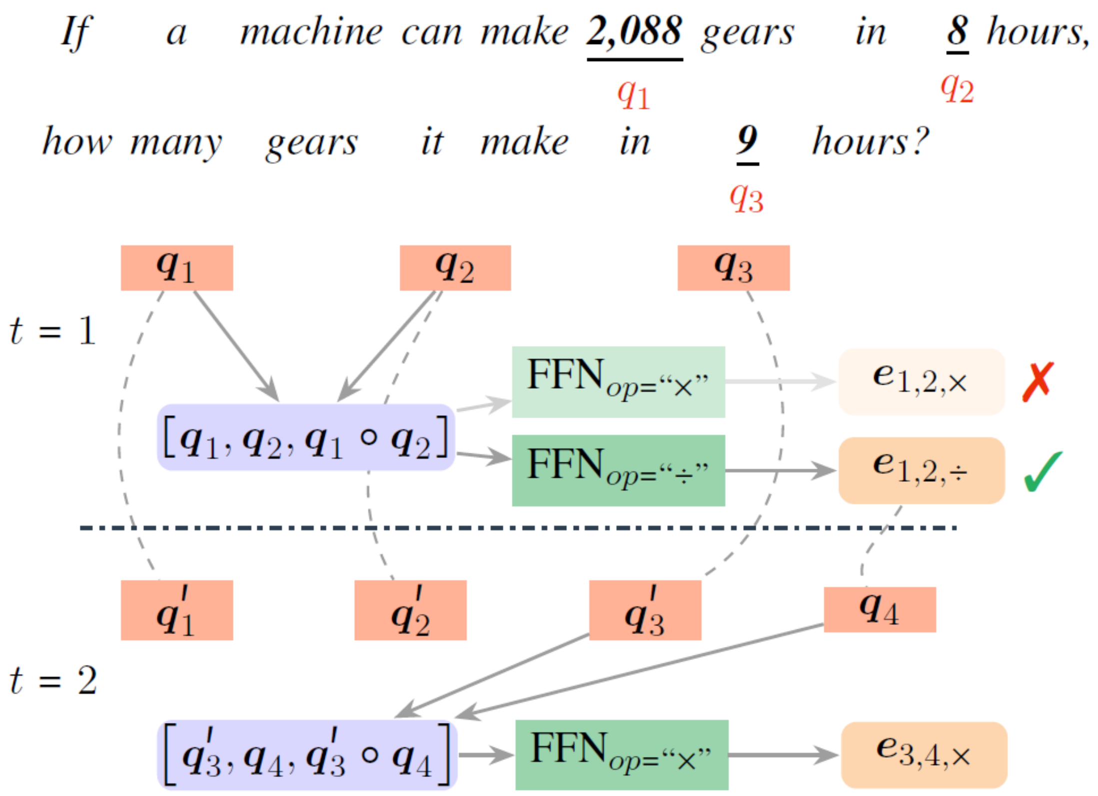
$$\boldsymbol{e}_{i,j,op} = \mathrm{FFN}_{op}([\boldsymbol{q}_i, \boldsymbol{q}_j, \boldsymbol{q}_i \circ \boldsymbol{q}_j]), \quad i \le j$$

  - $e^{(t)}_{i,j,op}$: The expression after applying the relation op to the ordered pair (q_i, q_j)
  - The constraint $i \le j$
    - Consider the “reverse operation” ("%"," − ")
    - The expression $\boldsymbol{e}_{1,2,\div}$ will be regarded as a new quantity with representation $\boldsymbol{q}_4$ at $t = 1$

Kenton Lee et al. End-to-end neural coreference resolution. EMNLP. 2017.

Part 5.

Model Components



Model architecture for the deductive reasoner

$$\text{“}q_1 \div q_2 \times q_3\text{”}$$

- **Reasoner**
  - Assign a score to a single reasoning step that yields the expression $e_{i,j,op}^{(t)}$

$$s(e_{i,j,op}^{(t)}) = s_q(\boldsymbol{q}_i) + s_q(\boldsymbol{q}_j) + s_e(\boldsymbol{e}_{i,j,op})$$

$$s_q(\boldsymbol{q}_i) = \mathbf{w}_q \cdot \text{FFN}(\boldsymbol{q}_i)$$

$$s_e(\boldsymbol{e}_{i,j,op}) = \mathbf{w}_e \cdot \boldsymbol{e}_{i,j,op}$$

  - Find the optimal expression sequence

$$[e^{(1)}, e^{(2)}, \cdots, e^{(T)}]$$

    - Enables us to compute the final numerical answer
    - T The total number of steps required for this deductive process

Model Components

- **Terminator**
  - A mechanism that decides whether the deductive procedure is ready to terminate at any given time
  - A binary label τ : 1 The procedure stops here, 0 otherwise

The final score of the expression e at time step t

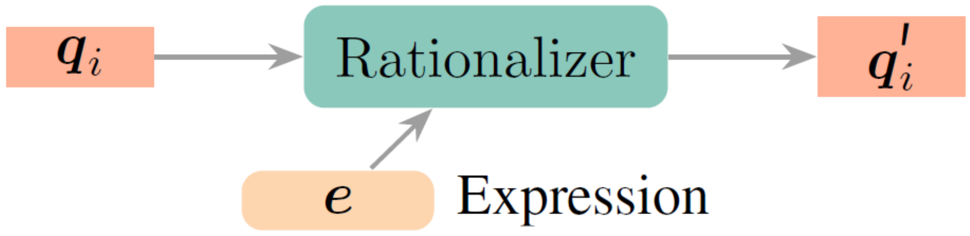
$$S(e^{(t)}_{i,j,op}, \tau) = s(e^{(t)}_{i,j,op}) + \mathbf{w}_{\tau} \cdot \mathrm{FFN}(\boldsymbol{e}_{i,j,op})$$

Model Components

- **Rationalizer**
  - Rationalization
    - Potentially give us the rationale that explains an outcome
    - Obtain a new intermediate expression at step t, it is crucial to update the representations for the existing quantities

  - If the quantity representations don't get updated with the deductive reasoning
  - Initially highly ranked expressions were (at the first step) would always be preferred over those lowly ranked ones throughout the process

Rationalizing quantity representation



- The intermediate expression e serves as the rationale that explains how the quantity changes from q to q'

$$q_i' = \text{Rationalizer}(q_i, e^{(t)}) \quad \forall\, 1 \le i \le |Q|$$

Model Components

- **Importance of Rationalizer**
  - If the quantity representations do not get updated as we continue the deductive reasoning process
  - Initially highly ranked expressions were (at the first step) would always be preferred over those lowly ranked ones throughout the process

- The first step is to predict $(1+2)*(3+4)$

- Intermediate expression $e^{(1)} = 1+2$

- The score of expression $s(e^{(1)}_{1,2,+}) > s(e^{(1)}_{3,4,+})$

- Without the rationalizer, representations for the quantities are unchanged $s(e^{(2)}_{1,2,+}) = s(e^{(1)}_{1,2,+}) > s(e^{(1)}_{3,4,+}) = s(e^{(2)}_{3,4,+})$

Model Components

- **Rationalizer**
  - Adopt well-known techniques as rationalizers
    - Allow us to update the quantity representation with the intermediate expression representation
  - Multi-head self-attention (Vaswani et al., 2017)
    - Construct a sentence with token representations (Quantity q_i & Previous expression e)
  - A gated recurrent unit (GRU) (Cho et al., 2014) cell
    - Use q_i as the input state and e as the previous hidden state in a GRU cell

The mechanism in different rationalizers

Rationalizer	Mechanism
Multi-head Self-Attention	$\text{Attention}(Q = [\boldsymbol{q}_i, \boldsymbol{e}], K = [\boldsymbol{q}_i, \boldsymbol{e}], V = [\boldsymbol{q}_i, \boldsymbol{e}])$
GRU cell	$\text{GRU\_Cell}(\text{input} = \boldsymbol{q}_i, \text{previous hidden} = \boldsymbol{e})$

Ashish Vaswani et al. Attention is all you need. NeurIPS. 2017.
Kyunghyun Cho et al. On the properties of neural machine translation: Encoder–decoder approaches. SSST-8, Eighth Workshop on Syntax. 2014.

Training & Inference

- **Training**
  - Adopt the teacher-forcing strategy (Williams and Zipser, 1989)
    - Similar to training sequence-to-sequence models (Luong et al., 2015)
    - Guide the model with gold expressions during training

Loss Function

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{t=1}^{T}\left(\max_{(i,j,op)\in\mathcal{H}^{(t)},\tau}\left[\mathcal{S}_{\boldsymbol{\theta}}(e_{i,j,op}^{(t)},\tau)\right] - \mathcal{S}_{\boldsymbol{\theta}}(e_{i^*,j^*,op^*}^{(t)},\tau^*)\right) + \lambda||\boldsymbol{\theta}||^2$$

$\boldsymbol{\theta}$ All parameters in the deductive reasoner

$\mathcal{H}^{(t)}$ All the possible choices of quantity pairs and relations available at time step t

λ The hyperparameter for the L_2 regularization term

Ronald J Williams et al. A learning algorithm for continually running fully recurrent neural networks. Neural computation. 1989.
Minh-Thang Luong et al. Effective approaches to attentionbased neural machine translation. EMNLP. 2015.

Training & Inference

- **Inference**
  - Set a maximum time step T_{max} and find the best expression e^* that has highest score at each time step
  - Once we see $\tau = 1$ is chosen, we stop constructing new expressions and terminate the process
  - The overall expression
    - It will formed by the resulting expression sequence
    - It will be used for computing the final numerical answer

Training & Inference

• Declarative Constraints

- Model repeatedly relies on existing quantities to construct new quantities
- Model results in a structure showing the deductive reasoning process

- It allows certain declarative knowledge to be conveniently incorporated

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{t=1}^{T}\left(\max_{(i,j,op)\in\mathcal{H}^{(t)},\tau}\left[\mathcal{S}_{\boldsymbol{\theta}}(e_{i,j,op}^{(t)},\tau)\right] - \mathcal{S}_{\boldsymbol{\theta}}(e_{i^*,j^*,op^*}^{(t)},\tau^*)\right) + \lambda||\boldsymbol{\theta}||^2$$

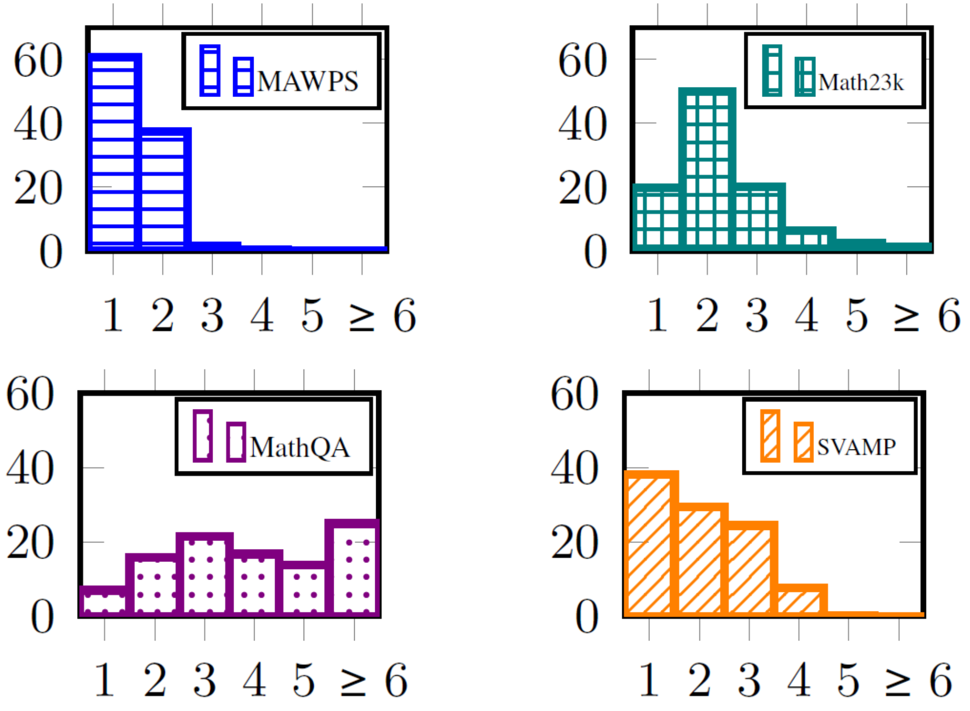
The default approach considers all the possible combinations among the quantities during the maximization step

- Easily impose constraints to avoid considering certain combinations
  - Find in certain datasets such as SVAMP, there does not exist any expression that involve operations applied to the same quantity ($9+9$, 9×9)
- Observe that the intermediate results would not be negative.
  - Simply exclude such cases in the maximization process, effectively reducing the search space during both training and inference

Part 7.

Experiments

- **Datasets**
  - MAWPS (Koncel-Kedziorski et al., 2016), SVAMP (Patel et al., 2021) ("+", " − ", " × ", " ÷ ")
  - Math23k (Wang et al., 2017), MathQA (Amini et al., 2019) ("+", " − ", " × ", " ÷ ", " ** ")
    - MathQA: Follow Tan et al. (2021) to adapt the dataset to filter out some questions that are unsolvable



Percentage of questions with different operation count

- MAWPS
  - 97% can be answered with only one or two operations
- MathQA
  - More than 60% have three or more operations
  - GRE questions in many domains including physics, geometry, probability, etc
- SVAMP
  - Variations from MAWPS: adding extra quantities, swapping the positions between noun phrases, etc.

Minghuan Tan et al. Investigating math word problems using pretrained multilingual language models. 2021.

Experiments

- **Baselines: Sequence-to-sequence (S2S)**
  - GroupAttn (Li et al., 2019)
    - Several types of attention mechanisms such as question or quantity related attentions
  - mBERT-LSTM(Lan et al. (2021))
    - Multilingual BERT with an LSTM decoder
  - BERT-BERT & Roberta-Roberta

- **Baselines: Sequence-to-tree (S2T)**
  - GTS (Xie and Sun, 2019)
    - Use a tree-based decoder with GRU
  - BERT-Tree (Liang et al., 2021; Li et al., 2021)
    - Use BERT as the encoder
  - NUMS2T (Wu et al., 2020) & NeuralSymbolic (Qin et al., 2021)
    - Solver incorporate external knowledge in the S2T architectures

- **Baselines: Graph-to-tree (G2T)**
  - Graph2Tree (Zhang et al., 2020)
    - Models the quantity relations using GCN

Experiments

- **Training Details**
  - Adopt BERT (Devlin et al., 2019) and Roberta (Liu et al., 2019) for the English datasets
  - Chinese BERT and Chinese Roberta (Cui et al., 2019) are used for Math23k
  - Use the GRU cell as the rationalizer
  - Also conduct experiments with multilingual BERT and XLMRoberta (Conneau et al., 2020)

  - Pre-trained models are initialized from HuggingFace's Transformers (Wolf et al., 2020)
  - Optimize the loss with the Adam optimizer
  - Use a learning rate of 2e-5 and a batch size of 30
  - The regularization coefficient λ is set to 0.01
  - Run our models with 5 random seeds and report the average results (with standard deviation)
  - Mainly report the value accuracy (percentage) in our experiments
  - 5-fold cross-validation results on both MAWPS8 and Math23k
  - The test set performance for Math23k, MathQA and SVAMP

Experiments

- **The Roberta encoder achieves the best performance**

Results on Math23k

	Model	Val Acc. Test	Val Acc. 5-fold
S2S	GroupAttn (Li et al., 2019)	69.5	66.9
	mBERT-LSTM (Tan et al., 2021)	75.1	-
	BERT-BERT (Lan et al., 2021)	-	76.6
	Roberta-Roberta (Lan et al., 2021)	-	76.9
S2T/G2T	GTS (Xie and Sun, 2019)	75.6	74.3
	KA-S2T† (Wu et al., 2020)	76.3	-
	MultiE&D (Shen and Jin, 2020)	78.4	76.9
	Graph2Tree (Zhang et al., 2020)	77.4	75.5
	NeuralSymbolic (Qin et al., 2021)	-	75.7
	NUMS2T† (Wu et al., 2021)	78.1	-
	HMS (Lin et al., 2021)	76.1	-
	BERT-Tree (Li et al., 2021)	82.4	-
OURS	BERT-DEDUCTREASONER	84.5 (± 0.16)	82.6 (± 0.17)
	ROBERTA-DEDUCTREASONER	**85.1** (± 0.24)	**83.0** (± 0.23)
	MBERT-DEDUCTREASONER	84.3 (± 0.19)	82.5 (± 0.33)
	XLM-R-DEDUCTREASONER	84.0 (± 0.22)	82.0 (± 0.12)

5-fold cross-validation results on MAWPS

	Model	Val Acc.
S2S	GroupAttn (Li et al., 2019)	76.1
	Transformer (Vaswani et al., 2017)	85.6
	BERT-BERT (Lan et al., 2021)	86.9
	Roberta-Roberta (Lan et al., 2021)	88.4
S2T/G2T	GTS (Xie and Sun, 2019)	82.6
	Graph2Tree (Zhang et al., 2020)	85.6
	Roberta-GTS (Patel et al., 2021)	88.5
	Roberta-Graph2Tree (Patel et al., 2021)	88.7
OURS	BERT-DEDUCTREASONER	91.2 (± 0.16)
	ROBERTA-DEDUCTREASONER	**92.0** (± 0.20)
	MBERT-DEDUCTREASONER	91.6 (± 0.13)
	XLM-R-DEDUCTREASONER	91.6 (± 0.11)

Part 7. Experiments

- **Performance with respect to different data splits, beam sizes**

Detailed comparison of different approaches on the Math23k dataset

Model	Beam Size	"Split Variant" Train/Val/Test 21162/1000/1000	Train/Test 22162/1000	5-fold CV	Customized Split	Remark
Group Attention (Li et al., 2019)	5	-	69.5	66.9	-	
GTS (Xie and Sun, 2019)	5	-	-	74.3	-	
KA-S2T (Wu et al., 2020)	5	-	-	-	76.3	They randomly split into 80%/20%
MultiE&D (Shen and Jin, 2020)	5	78.4 *(unknown)*†		76.9		
Graph2Tree (Zhang et al., 2020)	5	77.4		75.5	-	
NeuralSymbolic (Qin et al., 2021)	1	-	-	75.7	-	
NUM2ST (Wu et al., 2021)	5	-	-	-	78.1	They randomly split into train/val/test
HMS (Lin et al., 2021)	1	76.1	-	-	-	
BERT-Tree (Li et al., 2021)	3	82.4	-	-	-	
mBART-Large (Shen et al., 2021)	10	85.4 *(unknown)*†	†	84.3	-	
RoBERTa-DeductReasoner	1	84.3 (± 0.34)	86.0 (± 0.26)	83 (± 0.36)	-	
RoBERTa-Large-DeductReasoner	1	85.8 (± 0.42)	87.1 (± 0.21)	-	-	

Experiments

- **The Roberta encoder achieves the best performance**

Test accuracy comparison on MathQA

Model	Val Acc.
Graph2Tree (Zhang et al., 2020)	69.5
BERT-Tree (Li et al., 2021)	73.8
mBERT+LSTM (Tan et al., 2021)	77.1
BERT-DEDUCTREASONER	78.5 (± 0.07)
ROBERTA-DEDUCTREASONER	**78.6** (± 0.09)
MBERT-DEDUCTREASONER	78.2 (± 0.21)
XLM-R-DEDUCTREASONER	78.2 (± 0.11)

Test accuracy comparison on SVAMP

	Model	Val Acc.
S2S	GroupAttn (Li et al., 2019)	21.5
	BERT-BERT (Lan et al., 2021)	24.8
	Roberta-Roberta (Lan et al., 2021)	30.3
S2T/G2T	GTS* (Xie and Sun, 2019)	30.8
	Graph2Tree (Zhang et al., 2020)	36.5
	BERT-Tree (Li et al., 2021)	32.4
	Roberta-GTS (Patel et al., 2021)	41.0
	Roberta-Graph2Tree (Patel et al., 2021)	43.8
OURS	BERT-DEDUCTREASONER	35.3 (± 0.04)
	+ *constraints*	42.3 (± 0.09)
	ROBERTA-DEDUCTREASONER	45.0 (± 0.10)
	+ *constraints*	**47.3** (± 0.20)
	MBERT-DEDUCTREASONER	36.1 (± 0.07)
	+ *constraints*	41.3 (± 0.08)
	XLM-R-DEDUCTREASONER	38.1 (± 0.08)
	+ *constraints*	44.6 (± 0.15)

Additional Experiments *(experiments conducted after ACL conference)*	
All experiments are incorporated with constraints	
ROBERTA-DEDUCTREASONER†	48.9
DEBERTA-BASE-DEDUCTREASONER	55.6
DEBERTA-V3-LARGE-DEDUCTREASONER	62.0
DEBERTA-V2XX-LARGE-DEDUCTREASONER	63.6

Experiments

- **Accuracy under different number of operations**
  - Such comparisons on MathQA and SVAMP show that our model has a robust reasoning capability on more complex questions

#Operation	MAWPS		Math23k		MathQA		SVAMP	
	Baseline	OURS	Baseline	OURS	Baseline	OURS	Baseline	OURS
1	88.2	**92.7**	91.3	**93.6**	**77.3**	**77.4**	**51.9**	**52.0**
2	91.3	**91.6**	89.3	**92.0**	81.3	**83.5**	17.8	**32.1**
3	-	-	74.5	**77.0**	81.9	**83.4**	-	-
4	-	-	59.1	**60.3**	79.3	**81.7**	-	-
>=5	-	-	56.5	**69.2**	**71.5**	**71.4**	-	-
Overall Performance								
Equ Acc.	80.8	88.6	71.2	79.0	74.0	74.0	40.9	45.0
Val Acc.	88.7	92.0	82.4	85.1	77.1	78.6	43.8	47.3

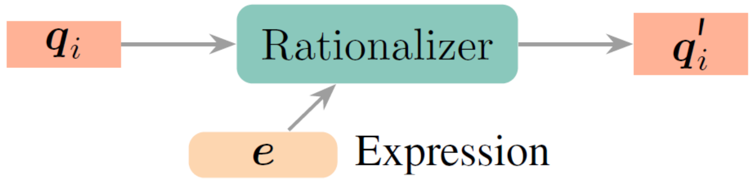
Experiments

- **Effect of Rationalizer**
  - The rationalizer is used to update the quantity representations at each step
  - So as to better "prepare them" for the subsequent reasoning process given the new context
  - GRU comes with sophisticated internal gating mechanisms
    - Internal gating mechanisms may allow richer representations for the quantities
  - The attention as a mechanism for measuring similarities (Katharopoulos et al., 2020)
    - It may be inherently biased when being used for updating quantity representations
    - When measuring the similarity between quantities and a specific expression, those quantities that have just participated in the construction of the expression may receive a higher degree of similarity

Performance comparison on different rationalizer using the Roberta-Base model

Rationalizer	MAWPS Equ Acc.	MAWPS Val Acc.	Math23k Equ Acc.	Math23k Val Acc.
NONE	88.4	91.8	71.5	77.8
Self-Attention	88.3	91.7	77.5	84.8
GRU unit	88.6	92.0	79.0	85.1

Rationalizing quantity representation



Angelos Katharopoulos et al. Transformers are rnns: Fast autoregressive transformers with linear attention. ICML. 2020.

Part 8.

Case Studies

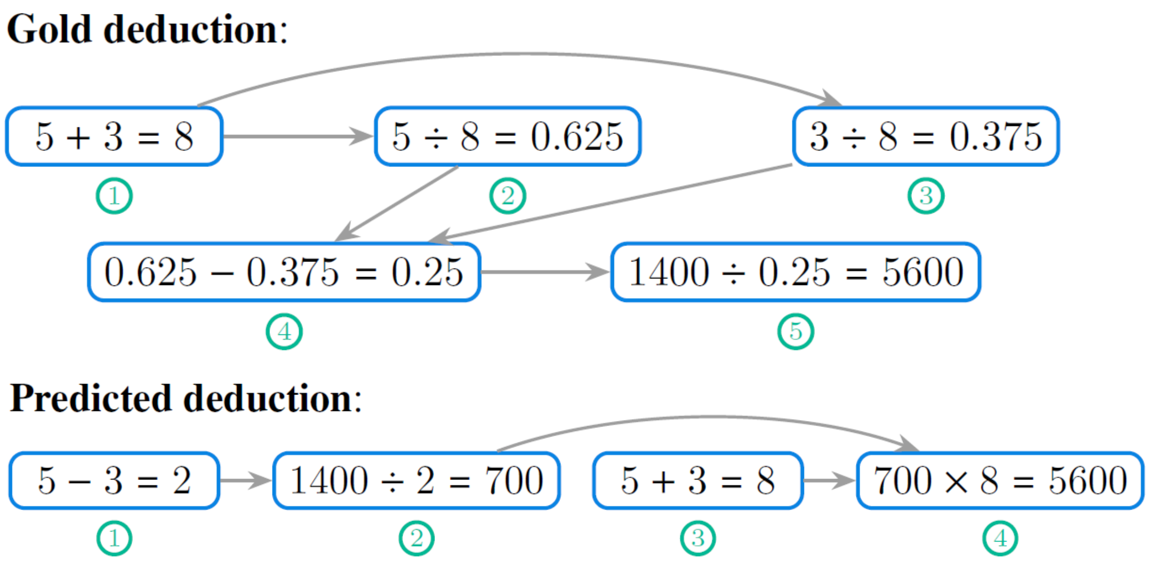
• Explainability of Output

- Our deductive reasoner is able to produce explainable steps to understand the answers
  - The predicted deductive process offers a slightly different understanding in speed difference

An example prediction from Math23k

Question: *Xiaoli and Xiaoqiang typed a manuscript together. Their typing speed ratio was* ***5:3****. Xiaoli typed* ***1,400*** *more words than Xiaoqiang. How many words are there in this manuscript?*

Gold Expr: $\frac{1400}{5\div(5+3)-3\div(5+3)}$ **Answer**: 5600



Case Studies

- **Question Perturbation**
  - Demonstrate the strong interpretability of our deductive reasoner
  - Highlights the important connection between math word problem solving and reading comprehension, a topic that has been studied in educational psychology (Vilenius-Tuohimaa et al., 2008)

An example prediction from Math23k

Question: *There are **255** apple trees in the orchard. Planting another **35** pear trees makes the number exactly the same as the apple trees. If every **20** pear trees are planted in a row, how many rows can be planted in total?*

Gold Expr: $(255 - 35) \div 20$ **Answer**: 11

Predicted Expr: $(255 + 35) \div 20$ **Predicted**: 14.5

Deductive Scores:

$\boxed{255 + 35 = 290}$ Prob.: 0.068 > $\boxed{255 - 35 = 220}$ Prob.: 0.062

Perturbed Question: *There are **255** apple trees in the orchard. **The number of pear trees is 35 fewer than the apple trees.** If every **20** pear trees are planted in a row, how many rows can be planted in total?*

$\boxed{255 + 35 = 290}$ **Prob.:** 0.061 < $\boxed{255 - 35 = 220}$ **Prob.:** 0.067

Practical Issues

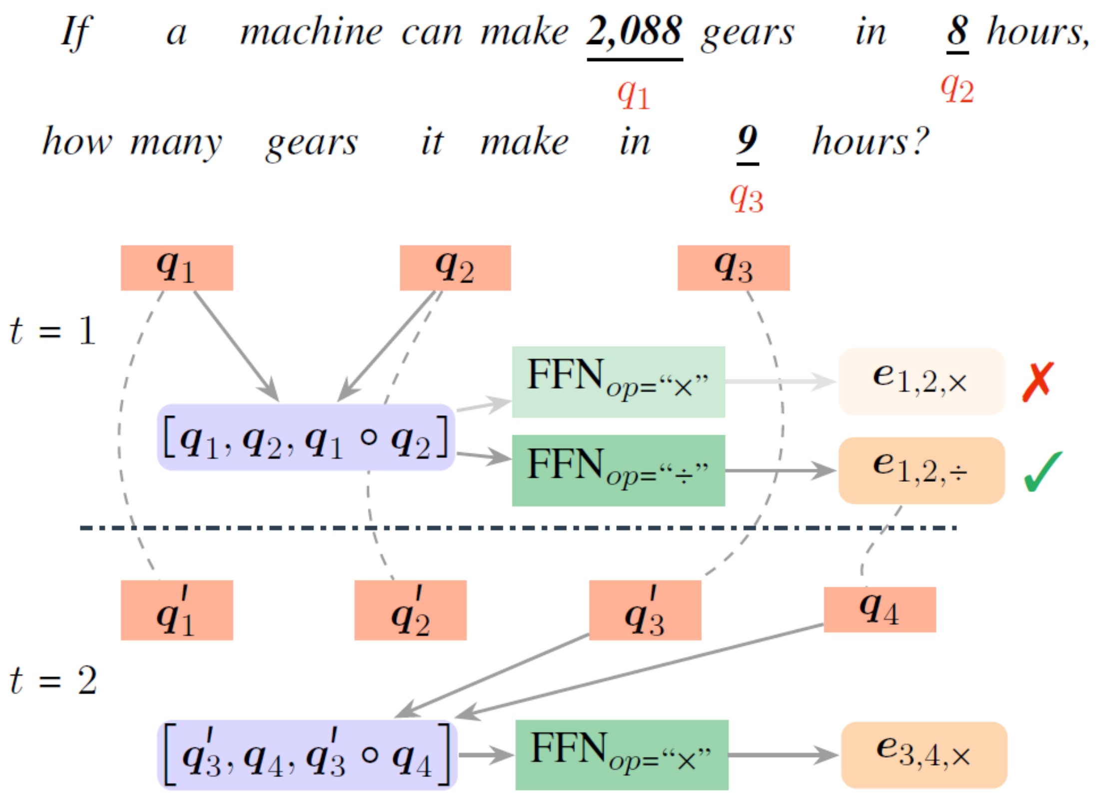
- **Assumption**
  - Needs to maintain a list of constants (e.g., 1 and π) as additional candidate quantities
  - Binary operators are considered
  - Beam search algorithm

Practical Issues

- **Needs to maintain a list of constants (e.g., 1 and π) as additional candidate quantities**
  - Select some top-scoring quantities & build expressions on top of them
  - However, a large number of quantities could lead to a large search space of expressions (i.e., $\mathcal{H}$)

Part 9.

Practical Issues



Model architecture for the deductive reasoner

$$\text{“}q_1 \div q_2 \times q_3\text{”}$$

- **Binary operators are considered**
  - Actually, extending it to support unary or ternary operators can be straightforward
  - Handling unary operators would require the introduction of some unary rules
  - A ternary operator can be defined as a composition of two binary operators

Practical Issues

- **Beam search algorithm**
  - One challenge with designing the beam search algorithm is that the search space $\mathcal{H}^{(t)}$ is expanding at each step t
  - Believe how to perform effective beam search in our setup could be an interesting research question that is worth exploring further.

Conclusion

- **An end-toend deductive reasoner**
  - Obtain the answer expression in a step-by-step manner
  - it can be fundamentally regarded as a complex relation extraction problem
    - At each step, our model performs iterative mathematical relation extraction between quantities
  - Achieve particularly better performance for complex questions that involve a larger number of operations
  - It offers us the flexibility in interpreting the results, thanks to the deductive nature of our mode

- **Future directions**
  - Explore include how to effectively incorporate commonsense knowledge into the deductive reasoning process, and how to facilitate counterfactual reasoning (Richards and Sanderson, 1999).

Part 3.

Method

$e_{i,j,op}^{(t)}$ q_i $Q^{(0)}$ $q_i \xrightarrow{op} q_j$ $e_{i,j,op}$ FFN_{op}

(q_i, q_j) L_2 $2e\text{-}5$

$$s_q(\boldsymbol{q}_i) = \mathbf{w}_q \cdot \mathrm{FFN}(\boldsymbol{q}_i)$$

Dataset	#Train	#Valid	#Test	Avg. Sent Len	#Const.	Lang.
MAWPS	1,589	199	199	30.3	17	English
Math23k	21,162	1,000	1,000	26.6	2	Chinese
MathQA†	16,191	2,411	1,605	39.6	24	English
SVAMP	3,138	-	1,000	34.7	17	English

T_{max} e^* $\boldsymbol{\theta} = \mathbf{0}$

"$5 \div (5+3) - 3 \div (5+3)$"

$(1400 \div 2)$ $225 - 35$ $e^{(1)}$ $e_{3,4,+}$