Using Intermediate Representations to Solve Math Word Problems

Danqing Huang[1], Jin-Ge Yao[2], Chin-Yew Lin[2], Qingyu Zhou[3], Jian Yin[1]
Sun Yat-sen University[1], Microsoft Research[2], Harbin Institute of Technology[3]

Microsoft Research 微软亚洲研究院

Math Problem Solving

The sum of 2 numbers is 18. The first number is 4 more than the second number. Find the two numbers.

Semantic gap between NL and equations

Learning target: x + y = 18, x = y + 4

What about "*the sum of 3/4/…/100 numbers*"?

Solution:

The sum of 2 numbers is 18. The first number is 4 more than the second number. Find the two numbers.

math#sum(cnt: 2) = 18, *ordinal*(1) = *math#more*(*ordinal*(2), 4)

x + y = 18, x = y + 4

"*the sum of 100 numbers*" → *math#sum*(cnt: 100)

Intermediate Representation

- **6 Classes**

int, *float*, *num*, *unk*, *var*, *list*

- **23 Math-Related Functions**

math#sum($1:list)
math#sum(cnt: $1:list)
ordinal($1:int)
….

Recursively applying joint operations with class type constraints
e.g. math#more(math#sum(cnt: 2), 4)

Learn Latent Presentation

- **Derive Latent Forms From Equations**

Regex/Rules	Class/Function
(<var>\+)+<var>	*math#sum*($1: list)
(<unk>\+)+<unk>	*math#sum*(cnt: $1:int)

The sum of twice a number and -21 is 129 greater than opposite of the number.
2 * m + (-21) = 129 + ((-1) * m)
→ *math#product*(2, m) + (-21) = 129 + *math#opposite*(m)
→ *math#sum*(*math#product*(2, m), -21) = *math#sum*(129, *math#opposite*(m))

- **Ambiguity in Derivation (~25% problems)**

Find 3 consecutive integers that 3 times the sum of the first and the third is 79.
3*(x+(x+2))=79

Possible LF derivations:
(1) math#consecutive(3), *math#product*(3, *math#sum*(*ordinal*(1), *ordinal*(3)) = 79
(2) *math#consecutive*(3), *math#product*(3, *math#sum*(*min*(), *max*()) = 79

- **Model: Seq2Seq with Attention Regularization**

$$Loss = -\sum_{i} \log p\left(y^i \middle| x^i; \theta\right) + \lambda * \left(\sum_{j=0}^{T} a_{ji} - 1\right)$$

Attention weights repetitively concentrate on "numbers"

- **Address Ambiguity: Iterative Labeling**

Algorithm 1 Iterative Labeling

Require:
(1) Tuples of (math problem description, equation system, answer) $D_n = \{(p_i, E_{p_i}, A_{p_i})\}$
(2) Possible latent forms $P_{LF} = \{(p_0, LF^1_{p_0}), (p_0, LF^2_{p_0}), ..., (p_n, LF^m_{p_n})\}$
(3) Beam size B
(4) training iterations N_{iter}, pre-training iterations N_{pre}
Procedure:
for $iter = 1$ to N_{iter} **do**
  if $iter < N_{pre}$ **then**
    $\theta \leftarrow$ MLE with P_{LF}
  else
    for (p, LF) in P_{LF} **do**
      C = Decode B latent forms given p
      for j in 1...B **do**
        if Ans(C_j) is correct **then**
          $LF \Leftarrow C_j$
          break
    $\theta \leftarrow$ MLE with relabeled P_{LF}

Initial training instances
$\{(p_0, LF^1_{p0}), (p_0, LF^2_{p0}), ..., (p_n, LF^m_{pn})\}$

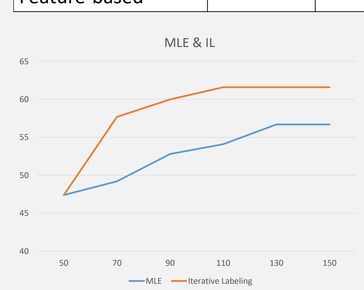
Model: seq2seq

Pre-train with all possible LF
(a more stable model)

Re-label LF in training data
Using IL, correct the LF of p_0 to LF^1_{p0}:
$\{(p_0, LF^1_{p0}), (p_0, LF^1_{p0}), ..., (p_n, LF^m_{pn})\}$

Experiment

Models	NumWord (Linear)	NumWord (ALL)	Dolphin18K (Linear)
s2s_Equ	26.8%	20.1%	13.1%
s2s_LF	50.8%	45.2%	13.9%
s2s_LF+AttnReg	56.7%	54.0%	15.1%
s2s_LF+AttnReg+Iter	**61.6%**	**57.1%**	**16.8%**
Shi et al., (2015) Rule-based	63.6%	60.2%	n/a
Huang et al., (2017) Feature-based	20.8%	n/a	28.4%



Iterative Labeling
- better performance
- faster convergence

- Effect of pretrain in Iterative Labeling

Iterative Labeling	NumWord (Linear)	NumWord (ALL)	Dolphin18K (Linear)
-pretrain	58.1%	54.9%	14.9%
+pretrain	61.6%	57.1%	16.8%

Neural? Rules? Features?

Two types of errors observed:

- **Natural language variations**

[x = 10 - 2]
"What is 10 minus 2?"
"John has 10 apples. How many apples does John have after giving Mary 2 apples."

- **Nested operations**

"I think of a number, double it, add 3, multiply the answer by 3 and then add on the original number."

SigmaDolphin in MSRA: www.microsoft.com/en-us/research/project/sigmadolphin-2