Learning To Use Formulas To Solve Arithmetic Problems

ARINDAM MITRA & CHITTA BARAL
DEPARTMENT OF COMPUTER SCIENCE,
ARIZONA STATE UNIVERSITY, USA

General Theme

- Answering questions (Q) with respect to natural language text (T) one often needs background knowledge. *How do we acquire such knowledge?*
- Even if we have a large collection of background knowledge only a small subset (S) of it may be relevant to T and Q. *How do we identify that subset S?*
- These concerns equally holds for the word arithmetic problem solving.

General Theme

- Most of us have experienced the role of examples in learning a new Math or Physics theory.
- The examples, help us to master which formula (that we have learned in the theory part) to use and how when solving a problem based on the theory.
- In this research we present a method that uses this approach to solve simple word arithmetic problems.

Math Formulas As Background Knowledge

- We have identified three kinds of “knowledge” (or “formulas”) that are involved in solving the problems in that collection.
- We model each formula as a template that has predefined slots and can be mapped to an equation when the slots are filled with numbers and unknown.
- Application of a formula to a problem is then defined as the instantiation of the template by a subset of quantities (known or unknown) from the problem.

Formula: Part-Whole

- The concept of part whole has two slots.
- One for the whole that accepts a single variable and the other for its parts that accepts a set of variables of size at least two.
- If the value of the whole is w and the value of the parts are $p_1, p_2, \dots, p_m$, then that application is mapped to the equation

$$w = p_1 + p_2 + \cdots + p_m$$

Formula: Change

- The change concept has four slots, namely *start*, *end*, *gains*, *losses* which respectively denote the original value of a variable, the final value of that variable, and the set of increments and decrements that happen to the original value of the variable.
- Given an instantiation of change concept the equation is generated as follows:

$$val_{start} + \sum_{g \in gains} val_g = val_{final} + \sum_{l \in losses} val_l$$

Formula: Comparison

- The comparison concept has three slots namely the *large quantity*, the *small quantity* and their *difference*.
- Each slot accepts single quantity. One of them must be the unknown.
- An instantiation of the comparison concept is mapped to the following equation:

$$large = small + difference$$

Example

Part-Whole	Change	Comparison
Dan grew 42 turnips and 38 cantaloupes. Jessica grew 47 turnips. How many turnips did they grow in total ?	Sam's dog had puppies and 8 had spots. He gave 2 to his friends . He now has 6 puppies. How many puppies did he have to start with?	Bill has 9 marbles. He has 7 more marbles than Jim. How many marbles does Jim have?
whole: x, parts: {42, 47}	start: x, losses: {2}, end: 6	large: 9, small: x, difference: 7
$x = 42 + 47$	$6 = x - 2$	$9 = x + 7$

Probabilistic Formulation

- Let A_P denotes the set of all possible applications to the problem P.
- Let P^1_{addsub} be set of all simple arithmetic problems which can be solved by a single application of a formula.
- Let H denotes the set $\{(P, y): P \in P^1_{addsub}, y \in A_P\}$.
- Let $\phi: H \rightarrow R^d$ denotes the feature function.

Probabilistic Formulation

- Given the definition of the feature function ϕ and the parameter vector θ the probability of an application y given a problem P is defined as:

$$p(y|P;\theta) = \frac{e^{\theta.\phi(P,y)}}{\sum_{y\prime \in A_p} e^{\theta.\phi(P,y\prime)}}$$

- The function f that computes the correct application is defined as:

$$f(P) = \arg\max_{y \in A_P} p(y|P;\theta)$$

Training

- We estimate θ by minimizing the negative of the conditional log-likelihood of the data:

$$O(\theta) = -\sum_{i=1}^{n} log\, p(y_i^*|P_i;\theta)$$
$$= -\sum_{i=1}^{n}[\theta.\phi(P_i, y_i^*) - log \sum_{y\in A_{P_i}} e^{\theta.\phi(P_i,y)}]$$

- We use stochastic gradient descent to optimize the parameters.

$$\frac{\nabla O}{\nabla \theta} = -\sum_{i=1}^{n}[\phi(P_i, y_i^*) - \sum_{y\in A_{P_i}} p(y|P_i;\theta) \times \phi(P_i, y)]$$

Overview of The Feature Function

Dan grew 42 turnips and 38 cantelopes . Jessica grew 47 turnips. How many turnips did they grow in total ?

Value	Type	Verb	HasAll
42	turnips	grow	false
38	cantelopes	grow	false
47	turnips	grow	false
x	turnips	grow	true

part-whole	whole: x, parts: {42, 47}
part-whole	whole: x, parts: {42}
part-whole	whole: x, parts: {42, 47, 38}

Activation Condition of Feature Dimensions
The *verb* of whole matches with the *verb* of parts.
The *type* of whole matches with the *type* of parts.
Whole has the *HasAll* attribute set to true.

(The value of each dimension when active is the number of variables attached to the formula, 0 otherwise.)

Case 1	**Case 2**	**Case 3**
3	2	3
3	2	1
3	2	1

Generated Feature Vectors

Experimental Evaluation

- The dataset consist of a total of 395 problems for third, fourth, and fifth graders. (Hosseini et.al.)
- The dataset is divided into three diverse set MA1, MA2, IXL containing 134, 140 and 121 problems respectively.
- Each problem in the dataset is annotated with the correct application of a formula.
- We evaluate our system with 3-fold cross validation.

Experimental Evaluation

	MA1	IXL	MA2	Avg
ARIS	83.6	75.0	74.4	77.7
KAZB	89.6	51.1	51.2	64.0
ALGES	-	-	-	77.0
Roy & Roth	-	-	-	78.0
Majority	45.5	71.4	23.7	48.9
Our System	96.27	82.14	79.33	86.07

Comparison with ARIS, KAZB (Kushman et al., 2014), ALGES (Koncel-Kedziorski et.al., 2015) and the state of the art Roy & Roth on the accuracy of solving arithmetic problems.

Experimental Evaluation

Type		MA1	IXL	MA2
Part-whole	Total	59	89	51
	Correct	59	81	40
Change	Total	74	18	68
	Correct	70	15	56
Compare	Total	0	33	0
	Correct	0	0	0

Accuracy on recognizing the correct application. None of the MA1 and MA2 dataset contains "compare" problems so the cross validation accuracy on "IXL" for "compare" problems is 0.

General Arithmetic Problems

Operators: {+, -, *, /}

New Formula (Unitary Formula)

If the cost of one object is given, we can find the cost of many objects by multiplying the cost of one object with the number of objects.

Unit Cost x Number of Objects=Total Cost

slot1 *slot 2* *slot 3*

Examples

- *Sara has saved 9 dollars from washing cars. How many dozen quarters does Sara have?*
- *After eating at the restaurant, Sally, Sam, and Alyssa decided to divide the bill evenly. If each person paid 45 dollars, what was the total of the bill?*
- *Benny bought a soft drink for 2 dollars and 5 candy bars. He spent a total of 27 dollars. How much did each candy bar cost?*
- *Oceanside Bike Rental Shop charges 17 dollars plus 7 dollars an hour for renting a bike. Tom paid 80 dollars to rent a bike. How many hours did he pay to have the bike checked out?*
- *Joey 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?*

Actions

- UseUnitConversionKnowledge
    - 1 dozen = 12
    - 1 dollar = 4 quarters
- Counting
    - Each person
    - They
- Join: $a_1 + a_2 + \ldots + a_m$
- Separate: $a - b$
- Multiply: $a * b$
- Divide: a/b

Problem Solving Algorithm

```
While true
    action = findNextAction(P);
    if(action is a Formula)
        //done
        break
    P.addQuantity(action.getResult());
End While
```