Supplementary Material for `Quantity Tagger`: A Latent-Variable Sequence Labeling Approach to Solving Addition-Subtraction Word Problems

Yanyan Zou and **Wei Lu**
StatNLP Research Group
Singapore University of Technology and Design
yanyan_zou@mymail.sutd.edu.sg, luwei@sutd.edu.sg

This is the supplementary material of Zou and Lu (2019).

1 Signs for Quantities

An instance of arithmetic word problem in the training set consists of a text described in natural language, an equation comprising constant quantities and the unknown x, as well as the solution to x, as shown in Figure 1. In order to obtain the *sign* for each quantity appearing in the problem text and for the unknown x associated to the question sentence, we apply a deterministic approach. First, we reconstruct the equation that is available in dataset by transposing the quantities located to the left side of the equal sign to the right-hand side, while maintaining the mathematical invariance. Exemplified by Problem 1 in Figure 1, the original equation provided by training data is $x = 7341 - 4221$. After transposition, we obtain the new equation $0 = 7341 - 4221 - x$. If a quantity is a minuend in this new equation, we deterministically consider the sign for such a quantity is "$-\mathbb{1}$", such as "4221" and x. Otherwise, the sign as "$+\mathbb{1}$, like "7341". If a constant quantity appears in the problem text but not in the equation, then the sign of this quantity is "$\mathbb{0}$". Therefore, for the quantity list $(2, 7341, 4221, x)$ of this problem, we can construct the corresponding sign list as $(\mathbb{0}, +\mathbb{1}, -\mathbb{1}, -\mathbb{1})$. Then, we can form an equation according to the quantity and sign list as "$(\mathbb{0})\times\mathbf{2} + (+\mathbb{1})\times\mathbf{7341} + (-\mathbb{1})\times\mathbf{4221} + (-\mathbb{1})\times x = 0$", which is mathematically equivalent to the original one.

2 Overlapped Quantity Spans

In this work, overlapped quantity spans are allowed. One token in $\boldsymbol{t}$ can belong to more than one quantity spans of different quantities.

Problem 1: *A worker at a medical lab is studying blood samples.* **2** *samples contained a total of* **7341** *blood cells. The first sample contained* **4221** *blood cells. How many blood cells were in the second sample?*
Prediction: $(\mathbb{0})\times\mathbf{2} + (+\mathbb{1})\times\mathbf{7341} + (-\mathbb{1})\times\mathbf{4221} + (-\mathbb{1})\times x = 0$ **Equation**: $7341 - 4221 - x = 0$ **Solution**: $x = 3120$
Problem 2: *There are* **22** *walnut trees currently in the park. Park workers will plant walnut trees today. When the workers are finished there will be* **55** *walnut trees in the park. How many walnut trees did the workers plant today?*
Prediction: $(+\mathbb{1})\times\mathbf{22} + (-\mathbb{1})\times\mathbf{55} + (+\mathbb{1})\times x = 0$ **Equation**: $22 - 55 + x = 0$ **Solution**: $x = 33$

Figure 1: Two examples of arithmetic word problems described in English with answers.

Text	Value	Type	Verb	Nsubj
Jason has ***43*** *blue and 16 red marbles.*	43	blue, marbles	has	Jason
Jason has 43 blue and ***16*** *red marbles.*	16	red, marbles	has	Jason
Tom has ***24*** *blue marbles.*	24	blue, marbles	has	Tom
How *many blue marbles do they have in all?*	x	blue, marbles	have	they

Table 1: Example quantities and their attributes.

3 Features

Features defined in $\mathbf{f}(\boldsymbol{t}, \boldsymbol{y}, \boldsymbol{h})$ are responsible for capturing information useful for modeling the sign sequence. We first define attributes for quantities that play a crucial role in constructing features, and then briefly introduce two categories of features used in this work. The rest of this section provides additional explanations for attributes and features exemplified by real problems from dataset.

Attributes. Exemplified by a real problem "*Jason has 43 blue and 16 red marbles. Tom has 24 blue marbles. How many blue marbles do they have in all?*", Table 1 lists available attributes of all constant quantities appearing in the text and the unknown quantity associated to the question sentence. For example, the sentence "*Jason has 43 blue and 16 red marbles.*" contains two constant quantities 43 and 16. Attributes of such two quantities are:

Contextual features
N-grams (N=1, 2, 3, 4), POS tags, output label transition patterns

Structural features
Are *types* of a constant quantity and the unknown the same?
Is the type of a constant quantity the subType of the unknown's?
Is the verb associated to a quantity the past tense?
Is the verb associated to a quantity the present tense?
Is the verb associated to a quantity the future tense?
Does the question sentence contain words in the "whole cue set"?
Does the question sentence contain comparative adjectives or comparative adverbs?
Is a constant quantity the largest?
Is a constant quantity the smallest?

Table 2: Features used in our models.

$\{\{$**value**:$\{43\}$, **type**: $\{$*blue, marbles*$\}$, **verb**:$\{$*has*$\}$, **nsubj**:$\{$*Jason*$\}\}$;
$\{$**value**:$\{16\}$, **type**: $\{$*red, marbles*$\}$, **verb**:$\{$*has*$\}$, **nsubj**:$\{$*Jason*$\}\}\}$.
According to the question sentence "*How many blue marbles do they have in all?*", it is obvious that the question is interested in "*blue marbles*". Hence the quantity $\mathbf{16}$ with *type* of "*red marbles*" is not relevant to the question. The system would assign a sign of "$\mathbb{0}$" to such a quantity with high probability.

Contextual Features. The contextual features capture quantity's context information. We extract *N-grams* ($N = 1, 2, 3, 4$) to gather locally contextual evidence surrounding a certain quantity. In practice, we replace a constant quantity with the token "number" and a unknown quantity with "x", instead of using the number/word appearing in the text. The N-gram features at one token/quantity position take the left/right word window in the size of N. The *POS tags* are generated by the Stanford Part-Of-Speech Tagger (Toutanova et al., 2003), which is beneficial to understanding the meaning of surrounding words. The *transition features* are defined on the output signs of adjacent quantities, which can capture the underlying transition patterns of mathematical equations.

Structural Features. Such structural features are capable of discovering the mathematical relations between quantities and questions. They also allow us to capture dependencies cross sentences.

One of main challenges for solving arithmetic word problems is to recognize irrelevant quantities which should be tagged with "$\mathbb{0}$". The attribute *type* of a quantity could be utilized to distinguish relevant and irrelevant quantities. For example, in Table 1, two quantities $\mathbf{43}, \mathbf{24}$ have the same *type* value $\{$*blue, marbles*$\}$ as the unknown, while $\mathbf{16}$ does not. From this observation, we can infer $\mathbf{16}$ is irrelevant to the question and is thus supposed to be tagged with "$\mathbb{0}$". Similarly, if there exists the *IsA* relation connecting types of two quantities in ConceptNet (Liu and Singh, 2004), then such two types form the *subType* relation, exemplified by the problem "*Alyssa bought some toys. She bought a football for \$5.71 , and spent \$6.59 on marbles . In total, how much did Alyssa spend on toys?*" where "football" and "marble" are subType of "toys". This implies that such two quantities are both relevant to the question.

The tense of *verb* words associated to quantities provides temporal information about the problem and is able to track how word states evolve over time. It also shows which time state the question is interested in. For instance, the problem "*There are 7 dogwood trees currently in the park. Park workers will plant 3 dogwood trees today and 2 dogwood trees tomorrow. How many dogwood trees will the park have when the workers are finished?*", the verb word in the question sentence is in future sense, and verb words for constant quantities either belong to present or future sense, which tells the machine that the question would be interested in both present and future status.

In practice, many of addition-subtraction problems contain *cue* words that imply addition or subtraction. We thus collect from training data and construct "whole cue set", containing "*all*", "*overall*", "*altogether*", "*together*", etc., for addition. Considering problem "*Before starting her shift, a waitress checks to make sure there is enough mustard for her customers. She finds 0.25 bottle at the first table, 0.25 bottle at the second table, and 0.375 bottle at the third table.* ***Altogether****, how many bottles of mustard does the waitress find?*", it is obvious that all relevant constant quantities should be summed together.

It is nature that if a question sentence contains a comparative adjective or adverb, such as "*longer*", "*more*", the question is more likely to be interested in the difference of two quantities. The prior is that the value of minuend should be larger than subtrahend. We thus add two more binary features to check if the quantity value is the largest or the smallest among all constant quantities, like one real problem "*Nicole found an orange caterpillar and a green caterpillar in her backyard. The green caterpillar was 3 inches long and the orange caterpillar was 1.1666666666666667 inches long. How much longer was the green caterpillar than the orange caterpillar?*".

Model	**AddSub**		**AS_CN**	
	J	$Acc.$	J	$Acc.$
QT(FIX)	3	87.81	3	42.57
QT	1	92.27	4	49.29
QT(S)	4	93.01	6	44.04
QT(R)	1	89.10	4	45.45

Table 3: Performance on development set. J: The length of window size of tokens; $Acc.$: Accuracy (%).

For all models, these structural features are called only at quantity positions, i.e., the current token refers to a quantity.

4 Additional Experiments

Hyperparameter J: To find the optimal hyperparameter J, we randomly select 80% instances of whole training set for training and the rest 20% for development. We consider $J \in \{1, 2, 3, 4, 5, 6, N\}$ and select the value of J based on development set. Note that all tokens of the problem text are selected as token window for a quantity when $J = N$. We report the optimal value of J for each model in Table 3 and the corresponding accuracies on the development set.

References

Hugo Liu and Push Singh. 2004. Conceptnet-a practical commonsense reasoning tool-kit. *BT technology journal*, 22.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of HLT-NAACL.*

Yanyan Zou and Wei Lu. 2019. Quantity Tagger: A latent variable sequence labeling approach to solving addition-subtraction word problems. In *Proc. of ACL.*