A Meaning-based English Math Word Problem Solver with Understanding, Reasoning and Explanation

Chao-Chun Liang, Shih-Hong Tsai, Ting-Yun Chang,
Yi-Chung Lin and Keh-Yih Su
Institute of Information Science, Academia Sinica, Taiwan
{ccliang,doublebite,terachang621,lyc,kysu}@iis.sinica.edu.tw

Abstract

This paper presents a meaning-based statistical math word problem (MWP) solver with understanding, reasoning and explanation. It comprises a web user interface and pipelined modules for analysing the text, transforming both body and question parts into their logic forms, and then performing inference on them. The associated context of each quantity is represented with proposed *role-tags* (e.g., nsubj, verb, etc.), which provides the flexibility for annotating the extracted math quantity with its associated syntactic and semantic information (which specifies the physical meaning of that quantity). Those role-tags are then used to identify the desired operands and filter out irrelevant quantities (so that the answer can be obtained precisely). Since the physical meaning of each quantity is *explicitly* represented with those role-tags and used in the inference process, the proposed approach could explain how the answer is obtained in a human comprehensible way.

1 Introduction

The *math word problem* (MWP) is frequently chosen to study natural language understanding for the following reasons: (1) The answer to the MWP cannot be simply extracted by performing keyword/pattern matching. It clearly shows the merit of understanding and inference. (2) An MWP usually possesses less complicated syntax and requires less amount of domain knowledge, so the researcher can focus on the task of understanding and reasoning. (3) The body part of MWP (which mentions the given information for solving the problem) consists of only a few sentences. The understanding and reasoning procedure thus could be checked more efficiently. (4) The MWP solver has its own applications such as *Computer Math Tutor* and *Helper for Math in Daily Life*.

According to the approaches used to identify entities, quantities, and to decide operands and operations, previous MWP solvers can be classified as: (1) Rule-based approaches (Mukherjee and Garain, 2008; Hosseini et al., 2014), which make all related decisions based on a set of rules; (2) Purely statistics-based approaches (Kushman et al., 2014; Roy et al., 2015), in which all related decisions are done via a statistical classifier; and (3) Mixed approach (Roy and Roth, 2015), which identifies entities and quantities with rules, yet, decides operands and operations via statistical classifiers.

The main problem of the rule-based approaches is that a wide coverage rule-set is difficult and expensive to construct. Also, it is awkward in resolving ambiguity problem. In contrast, the main problems of the purely statistics-based approaches are that the performance deteriorates significantly when the MWP is complicated, and they are sensitive to the irrelevant information (Hosseini et al., 2014).

A meaning-based[1] statistical framework (Lin et al., 2015) is thus proposed to perform understanding and reasoning to avoid the problems mentioned above. The proposed *role-tags* (e.g., nsubj, verb, etc.) provides the flexibility for annotating extracted math quantities with their associated syntactic and se-

[1] According to the study reported by Pape (2004), the meaning-based approach for solving MWPs achieves the best performance among various behaviours adopted by middle school children.

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations, pages 151–155, Osaka, Japan, December 11-17 2016.

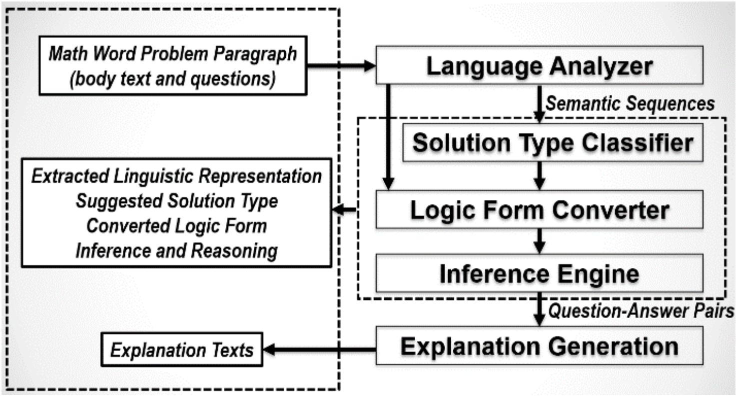

Figure 1: The block diagram of the MWP Solver

semantic (such as co-reference) information (in the context), which can be used to identify the desired operand, filter out irrelevant and perform inference to solve MWPs.

2 System Architecture

The block diagram of our English MWP solver is shown in Figure 1 (Lin et al., 2015). The sentences in a MWP are analyzed by the *Language Analyzer* (LA) module (i.e., Stanford CoreNLP suite (Manning et al., 2014)) to obtain corresponding linguistic representation (i.e., dependency trees and co-reference chains). Then, the *Solution Type Classifier* (STC), which is an SVM classifier adopting linear kernel functions, determines the solution type for each question in the MWP. According to the given solution type, the *Logic Form Converter* (LFC) transforms the linguistic representation into logic forms. Afterwards, based on the logic forms, the *Inference Engine* (IE) generates the answer for each question. Finally, the *Explanation Generator* (EG) module generates the explanation text to explain how the answer is obtained according to the given reasoning chain (Russell and Norvig, 2009).

2.1 Solution Type Identification

The solution type is the key math operation to solve a question in an MWP. In the classroom, children are usually taught with various MWPs of the same solution type, such as addition, multiplication, greatest common divisor, and so on. Teaching the MWPs of the same solution type at a time is helpful for learning because they share the similar patterns (in language usages or in logic representations/inferences). Once the solution type of an MWP is identified, solving the MWP becomes easier. Based on this strategy, the STC is adopted in our system to identify the solution type of MWPs.

The STC will select a math operation (that LFC should adopt to solve the problem) based on the global information across various input sentences. We classify the English MWPs into 6 main solution types: "*Addition*", "*Subtraction*", "*Multiplication*", "*Division*", "*Sum*" and "*TVQ*". The first five types are self-explained with their names. The last one "*TVQ*" means to get the initial/change/final value of a specific *Time-Variant-Quantity*. Currently, an SVM classifier with linear kernel functions (Chang and Lin, 2011) is used, and it adopts three different kinds of feature-sets: (1) Verb Category (Bakman, 2007; Hosseini et al., 2014) related features, (2) various *key-word indicators* (such as "*total*" and "*in all*" which frequently indicate an addition operation), and (3) indicators for various specified aggregative patterns (e.g. "*If the Body contains only two quantities, and their associated verbs are the same*" which frequently implies the "*Addition*" solution type).

2.2 Logical Form Transformation

A two-stage approach is adopted to transform the linguistic representation into logic forms for solving MWPs. In the first stage, the FOL predicates are generated by traversing the input linguistic representation. For example, "*Fred picked 36 roses.*" will be transformed into the following FOL predicates separated by the logic AND operator "*&*" and the first arguments, *v1*, *n1* and *n2*, are the identifiers.

verb(v1,pack)&nsubj(v1,n1)&dobj(v1,n2)&head(n2,rose)&nummod(n2,36)

In the second stage, crucial generic math facts associated with quantities and relations between quantities are generated. For example, the FOL function "*quan(q_{id},unit,object)=number*" is used to describe the facts about quantities. The first argument is a unique identifier to represent the quantity fact. The other arguments and the function value describe the meaning of this fact. For the above example, a quantity fact "*quan(q1,#,rose)=36*" is generated. Auxiliary domain-independent facts associated with domain-dependent facts like *quan(. . .)* are also created in this stage to help the IE find the solution. For example, the auxiliary fact "*verb(q1,pick)*" is created for *q1* to state "*the verb of q1 is pick*".

The FOL predicate "*qmap*(map_{id},q_{id1},q_{id2})", which denotes the mapping from q_{id1} to q_{id2}, is used to describe a relation between two quantity facts, where the first argument is a unique identifier to represent this relation. For example, *qmap(m1,q3,q4)* indicates that there is a relation between "100 candies" (*quan(q3,#,candidate)=100)* and "5 boxes" (*quan(q4,#,box)=5)* in the example of "*Pack 100 candies into 5 boxes*". The auxiliary fact "*verb(m1,pack)*" is created for *m1* to state "*the verb of m1 is pack*".

The questions in an MWP are transformed into FOL-like utility functions provided by the IE according to the suggested solution type. One utility function is issued for each question to find the answer. According to the solution type provided by the STC, the LFC will select an IE utility and instantiate its arguments. For example, if "*How many roses were picked in total?*" is labelled with "*Sum*" by the STC, the LFC will transform it to "*ASK Sum(quan(?q,rose),verb(?q,pick))*", which asks the IE to sum the values of all quantity facts of which verbs are "*pick*".

2.3 Logic Inference

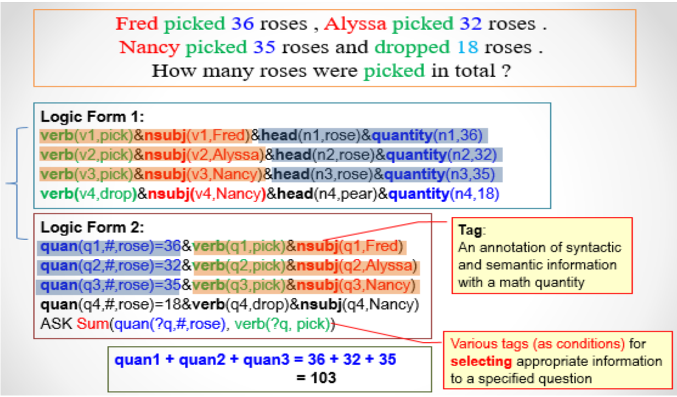

Figure 2: Logic form and logic inference of a *Sum* operation

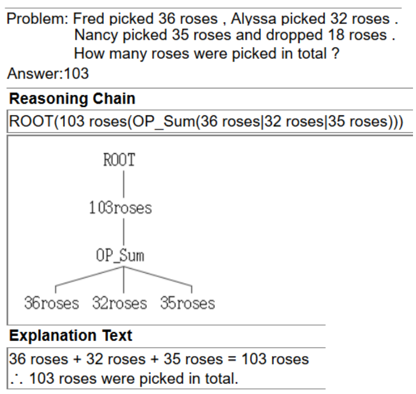

Figure3: The generated explanation tree and explanation text

The IE is used to find the solution of an MWP. Currently, IE provides 9 different utilities to perform simple arithmetical operations. It is responsible for providing utilities to select desired facts and then obtain the answer by taking math operations on those selected facts. For example, the Addition utility, "*Addition*($value_1$, $value_2$) = *value*", returns the value of "$value_1$+$value_2$", where $value_i$ could be a constant number, an FOL function value, or a value returned by a utility; and the Sum utility, "*Sum(function , condition)*", returns the sum of the values of FOL function instances which can be unified with the function arguments and satisfy the condition arguments. IE is also responsible for using inference rules to derive new facts from those facts which are directly derived from the description of the MWP. Consider the example shown in Figure 2, the IE will first select all qualified quantities which match "*quan(?q, #, rose)*" and with a "*pick*" verb-tag, and then performs a "*Sum*" operation on them. The irrelevant quantity "*quan(q4, #, rose)*" in that example is pruned out as its verb-tag is "*drop*", not "*pick*". The answer is then obtained by summing those quantities *q1*, *q2* and *q3*.

2.4 Explanation Generation

The EG is responsible for explaining the associated reasoning steps in fluent natural language based on the reasoning chain generated from IE. A math operation oriented approach (Huang et al., 2015) is adopted to explain how the answer is obtained. It first converts the given reasoning chain into its corresponding *Explanation Tree*, which represents the associated operations and operands for solving the MWP. After that, a specific template is used to generate the explanation text for each kind of operation. Consider the example shown in Figure 3, the explanation text "*36 roses + 32 roses + 35 roses = 103 roses. ∴103 roses were picked in total.*" will be generated to explain that the obtained answer is a summation of "*36 roses*", "*32 roses*" and "*35 roses*".

3 Experiments

	MA1	IXL	MA2	Total
3-fold Cross validation				
Our System	**94.8**	73.4	**88.4**	**85.3**
UIUC	-	-	-	78.0
ARIS	83.6	**75.0**	74.4	77.7
KAZB	89.6	51.1	51.2	64.0
Gold Solution Type				
Our System	99.3	97.8	95.0	97.5
STC accuracy	91.8	74.1	79.6	81.7

Table 1: Accuracy rates of different systems in AI2-395. "Total" denotes the micro-average performance. "Gold Solution Type" reports the accuracy from the gold solution type.

	IL-562
5-fold Cross validation	
Our System	**79.5**
UIUC	73.9
ARIS	-
KAZB	73.7

Table 2: Accuracy rates of different systems in IL-562 dataset.

We evaluate our system on two publicly available datasets, *AI2-395* and *IL-562*. **AI2-395** includes 395 *Addition* and *Subtraction* MWPs which are provided by Hosseini et al. (2014). It includes three sub-datasets (i.e., MA1, IXL and MA2) with different feature categories. **IL-562** is a collection of 562 arithmetic word problems released by Roy et al. (2015), and each of them can be solved with only one math operation among *Addition*, *Subtraction*, *Multiplication* or *Division*.

We compare our system with the rule-based approach ARIS (Hosseini et al., 2014), the purely statistical approach KAZB (Kushman et al., 2014), and the mixed approach UIUC system (Roy and Roth, 2015). We follow the same evaluation setting adopted in (Hosseini et al., 2014) and (Roy et. al, 2015). Table 1 and 2 show that our system significantly outperform theirs in overall performance.

4 Demonstration Outline

The MWP solver comprises a web user interface (Figure 4) and a processing server. The web interface is used to input the problem and display various outputs generated from the submitted MWP. The server will process the submitted problem to get the answer. After an MWP is submitted, various processing modules will be invoked in a pipelined manner (Figure 1) to solve the problem. Once the process is finished, the user can browse the outputs generated from each module: (1) Corresponding *dependency relations*, *co-reference chains* and *linguistic representations*, which are generated from *LA*. (2) Suggested *solution type*, which identifies the desired math operation. (3) Obtained *logical forms*, which are transformed from the linguistic representation and the specified solution type. (4) Generated *reasoning chain* and *explanation text* (Figure 3), which explains how the problem is solved. An online demo is available at: http://nlul.iis.sinica.edu.tw/EnglishMathSolver/mathDemo.py.

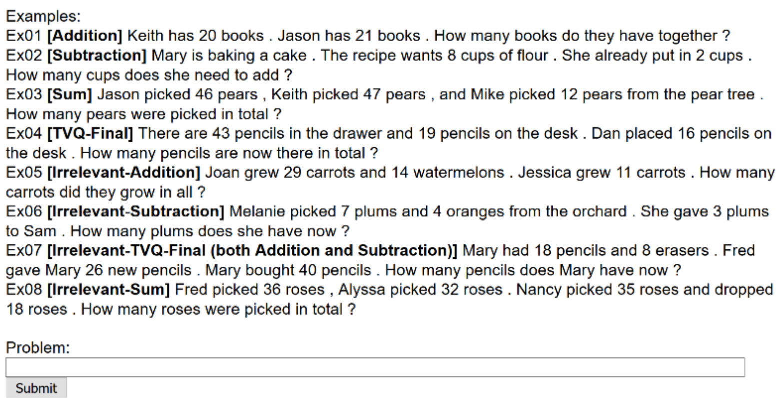

Figure 4: A web interface of the MWP Solver

5 Conclusion

A meaning-based logic form represented with role-tags is first proposed to provide the flexibility for annotating the extracted math quantity with its associated syntactic and semantic information in the context. Those tags can be used to identify the desired operands and filter out irrelevant quantities. Since the physical meaning of each quantity is *explicitly* expressed and used during inference, the associated reasoning procedure is human comprehensible and could be easily explained to the user.

A statistical framework based on the above meaning-based logic form is then proposed in this paper to perform understanding and reasoning for solving the given MWP. The combination of the statistical framework and logic inference distinguishes the proposed approach from other approaches.

The main contributions of our work are: (1) Proposing a meaning-based logic representation so that the physical meaning of each quantity could be explicitly specified and used in getting the answer; (2) Proposing a statistical framework for performing reasoning from the given MWP text.

6 Future Works

Currently, the MWP solver assumes that the final answer can be directly obtained from those known quantity facts via only one arithmetic operation (i.e., merely handling one-step MWPs). It cannot solve the problem if multiple arithmetic operations are required. For example, "*Mary had 92 pieces of candy. She gave 4 pieces each to 9 friends. How many pieces of candy does Mary have left?*" is not handled now. A goal oriented approach for handling the above multi-step MWP is thus proposed and under test. Besides, the current system cannot handle some subtle referring relationships. For instance, the system does not know that "*customers*" refers to "*women*" and "*men*" in the following MWP "*A waiter had 6 tables he was waiting on, with 3 women and 5 men at each table. How many customers total did the waiter have?*". Advanced analysis is required to solve this kind of problems.

Reference

Yefim Bakman. 2007. Robust understanding of word problems with extraneous information. http://lanl.arxiv.org/abs/math.GM/0701393.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27. At http://www.csie.ntu.edu.tw/cjlin/libsvm.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. *EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 523–533.

Chien-Tsung Huang, Yi-Chung Lin, and Keh-Yih Su. 2015. Explanation generation for a math word problem solver. *IJCLCLP*, 20(2):27–44.

Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. 2014. Learning to automatically solve algebra word problems. Baltimore, Maryland, June. *Association for Computational Linguistics*, page 271–281.

Yi-Chung Lin, Chao-Chun Liang, Kuang-Yi Hsu, Chien-Tsung Huang, Shen-Yun Miao, Wei-Yun Ma, Lun-Wei Ku, Churn-Jung Liau, and Keh-Yih Su. 2015. Designing a tag-based statistical math word problem solver with reasoning and explanation. *IJCLCLP*, 20(2):1–26.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL Demonstrations*.

Anirban Mukherjee and Utpal Garain. 2008. A review of methods for automatic understanding of natural language mathematical problems. *Artif Intell Rev*, 29(2):93–122.

Stephen J. Pape. 2004. Middle school children's problem-solving behavior: A cognitive analysis from a reading comprehension perspective. *Journal for Research in Mathematics Education*, 35(3):187–219.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. *EMNLP*, pages 1743–1752.

Subhro Roy, Tim Vieira, and Dan Roth. 2015. Reasoning about quantities in natural language. *TACL*, 3:1–13.

Stuart Russell and Peter Norvig. 2009. *Artificial Intelligence*: A Modern Approach. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition.