A Goal-Oriented Meaning-based Statistical Multi-Step Math Word Problem Solver with Understanding, Reasoning and Explanation (Demonstration)

Chao-Chun Liang, Yu-Shiang Wong, Yi-Chung Lin and Keh-Yih Su
Institute of Information Science, Academia Sinica, Taiwan
{ccliang, yushiangwtw, lyc, kysu}@iis.sinica.edu.tw

Abstract

A *goal-oriented meaning-based* statistical framework is presented in this paper to solve the math word problem that requires multiple arithmetic operations with understanding, reasoning and explanation. It first analyzes and transforms sentences into their *meaning-based* logical forms, which represent the associated context of each quantity with *role-tags* (e.g., *nsubj, verb*, etc.). Logic forms with role-tags provide a flexible and simple way to specify the physical meaning of a quantity. Afterwards, the main-goal of the problem is decomposed recursively into its associated sub-goals. For each given sub-goal, the associated operator and operands are selected with statistical models. Lastly, it performs inference on logic expressions to get the answer and explains how the answer is obtained in a human comprehensible way. This process thus resembles the human cognitive understanding of the problem and produces a more meaningful problem solving interpretation.

1 Introduction

The math word problem (MWP) was frequently chosen to study the task of natural language understanding and simulate human problem solving procedure [Bakman, 2007; Kushman *et al*., 2014; Liang *et al.*, 2016]. However, many previous approaches either only handled MWPs that involve one arithmetic operation [Mukherjee and Garain, 2008; Roy *et al.*, 2015] or solved multi-step arithmetic operations with various limitations (e.g., only handling *Addition* and *Subtraction* sequence [Ma *et al.*, 2010; Hosseini *et al.*, 2014; Mitra and Baral, 2016]). Only [Roy and Roth, 2015] and [Koncel-Kedziorski *et al.*, 2015] proposed bottom-up tree construction approaches to handle general multi-step arithmetic MWPs. But both of them lacked intermediate understanding and explanation. It thus belongs to the worse *direct translation approach* [Pape, 2004]. In contrary, previous *goal-oriented* approaches [Slagle, 1965; Ma *et al*., 2010] largely aligned with the human comprehension process, though they could not handle general cases and suffered from the difficulty of constructing a wide coverage *rule-set*.

A *goal-oriented meaning-based* statistical approach is thus proposed in this paper to avoid the problems mentioned above via combining the *goal-oriented* approach with a *statistical* framework. Except the ultimate goal, all the desired operations and operands (including sub-goals) will be identified by statistical classifiers. Moreover, each quantity is associated with various role-tags (e.g., *nsubj*, *verb*, *modifier*, *time*, *place,* etc.) for denoting its contextual relationship. Since the physical meaning (represented by role-tags) of each quantity is explicitly used in selecting both operations and operands, it allows us to examine the problem in an incremental and intuitive manner. In each step of the process, we know the meaning of the associated sub-goals/operands, and are thus able to explain the procedure in a human-comprehensible way [Mayer, 1987; 1992].

2 System Architecture

The block diagram of the proposed MWP solver is shown in Figure 1. The sentences in an MWP are first analyzed by the *Language Analyzer* (LA) (i.e., Stanford CoreNLP suite [Manning *et al.*, 2014]) to obtain corresponding linguistic representation (i.e., dependency trees and co-reference chains). The *Logic Form Converter* (LFC) then transforms the linguistic representation into logic forms and constructs the final operation tree. Besides, the LFC also calls the *Solution Type Classifier* (STC) to determine the solution type and calls the *Inference Engine* (IE) [Liang *et al.*, 2016] to evaluate the given operation tree and generate the answer for the question. Lastly, the *Explanation Generator* (EG) [Huang *et al.*, 2015] generates the explanation text to explain how the answer is obtained according to the given reasoning chain.

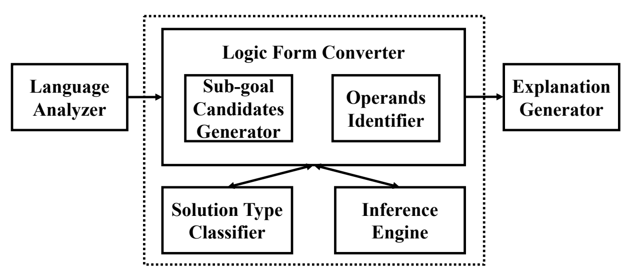


Figure 1: The block diagram of the proposed MWP Solver

We adopt the same STC, IE and EG modules used in [Liang *et al.*, 2016], and only describe new modules below.

2.1 Logical Form Converter

The LFC transforms each quantity into its *first-order logic form* [Lin *et al.*, 2015; Russell and Norvig, 2009]. All quantities and the *main-goal* are first identified by rules and explicitly associated with their *role-tags*, which carries context information and specifies the physical meaning of that quantity. For example, "*quan(q1,#,apple)=3 & verb(q1, eat) & nsub(q1, Mary)*" is generated to state '*Mary ate 3 apples*'. In contrast with the *bottom up* approach adopted in [Roy and Roth, 2015; Koncel-Kedziorski *et al.*, 2015], the LFC constructs the operation tree [Roy and Roth, 2015] in a *top-down* manner. It initializes a *recursive procedure* by first pushing the main-goal into a stack. Afterwards, it pops the current working goal from the stack, and then asks the STC to decide the desired solution type (i.e., the operator) associated with the goal. It then generates all possible candidates of sub-goals (via the rule-based *Sub-goal Candidates Generator* sub-module), and selects associated operands (via the statistical *Operands Identifier* sub-module) among all available *known* quantities and those *sub-goal candidates* just generated. Afterwards, it spawns a new level of recursion if the selected operand is an *unknown* quantity (i.e., a sub-goal candidate generated above), which will be regarded as a new sub-goal and pushed into the stack. The above procedure will keep going until the stack is empty. Lastly, the LFC identifies the corresponding IE utility (associated with the solution type), and calls the IE to get the answer.

2.2 Operands Identifier

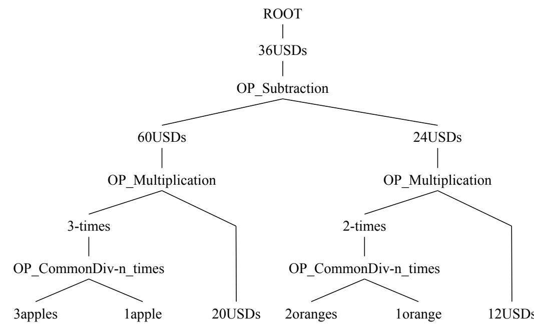
The *Operand Identifier* selects the appropriate operands from the *Operand-Candidate-Set*, which is a set formed by newly generated sub-goals and currently available known quantities. An SVM classifier with linear kernel [Chang and Lin, 2011] is used. We adopt three different kinds of features: (1) *Math fact pattern features* (e.g., "if the operand is a known quantity", etc.); (2) *NP-matching features* (e.g., "if the noun-phrases of the operand-candidate and the goal are in entailment relationship", etc.); (3) *Role-tag-matching features* (e.g. "if the role-tags of operand-candidate and that of goal are matched", etc.).

3 Experimental Results

We evaluate our system on two publicly available datasets: IL-562 and CC-600. IL-562 is a collection of 562 MWPs (with single arithmetic operation) released by [Roy *et al.*, 2015]. CC-600 is a dataset of 600 MWPs released by [Roy and Roth, 2015] to cover multi-step MWPs with four different arithmetic operations. We compare our system with the KAZB [Kushman *et al.*, 2014] and the system proposed by [Roy and Roth, 2015]. Since exactly the same n-fold cross-validation evaluation setting is adopted, the performances can be directly compared. Table 1 shows that our system significantly outperforms theirs in overall performance. We believe the improvement is mainly due to that the system explicitly checks the physical meaning of the selected quantities against the meaning of the given sub-goal.

	IL-562	CC-600
Our System	**80.1**	**53.5**
Roy and Roth, 2015	73.9	45.2
Kushman *et al.*, 2014	73.7	2.3

Table 1: Accuracy Comparison



Question: An apple is priced at $ 20 and an orange is priced at $ 12 . How much more do 3 apples cost than 2 oranges ?
Answer and Explanation:
3 apples vs. 1 apple is: 3 ÷ 1 = 3-times
3-times and 20 USDs is: 3 * 20 = 60 USDs
2 oranges vs. 1 orange is: 2 ÷ 1 = 2-times
2-times and 12 USDs is: 2 * 12 = 24 USDs
60 USDs - 24 USDs = 36 USDs
∴ 3 apples cost 36 USDs more expensive than 2 oranges.

Figure 2: The reasoning chain and explanation text

4 Demonstration Outline

The MWP solver comprises a web user interface 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 recursion manner to solve the problem. Once the process is finished, the user can browse the outputs generated from each module: (1) Dependency relations, co-reference chains and linguistic representations generated from the LA. (2) Logic forms transformed from the linguistic representation and the specified solution type. (3) Reasoning chain and explanation text (Figure 2), which explains how the problem is solved. An online demo is available via the following web address: http://nlul.iis.sinica.edu.tw/EnglishMathSolver/mathDemoMS.py.

5 Conclusion

A goal-oriented meaning-based statistical framework is proposed to solve multi-step MWPs in a top-down, recursive manner. The approach resembles the human cognitive understanding of math word problems, and thus allows us to provide an intuitive human-comprehensible explanation to the problem-solving process.

References

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

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

[Hosseini *et al.*, 2014] Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 523–533, *October 25-29, 2014.*

[Huang *et al.*, 2015] Chien-Tsung Huang, Yi-Chung Lin, and Keh-Yih Su. Explanation generation for a math word problem solver. *International Journal of computational Linguistics and Chinese Language processing (IJ-CLCLP)*, 20(2):27–44, 2015.

[Koncel-Kedziorski *et al.*, 2015] Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, 3:585–597, 2015.

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

[Liang *et al.*, 2016] Chao-Chun Liang, Shih-Hong Tsai, Ting-Yun Chang, Yi-Chung Lin and Keh-Yih Su. A Meaning-based English Math Word Problem Solver with Understanding, Reasoning and Explanation. *Proceedings of COLING 2016*, 2016.

[Lin *et al.*, 2015] 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. Designing a tag-based statistical math word problem solver with reasoning and explanation. *International Journal of computational Linguistics and Chinese Language processing (IJCLCLP)*, 20(2):1–26, 2015.

[Ma *et al.*, 2010] Yuhui Ma, Ying Zhou, Guangzuo Cui, Yun Ren, and Ronghuai Huang. Frame-based calculus of solving arithmetic multi-step addition and subtraction word problems. *Education Technology and Computer Science, International Workshop on*, 2:476–479, 2010.

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

[Mayer, 1987] Richard E. Mayer. Educational psychology: A cognitive approach. Boston: Little, Brown, 1987.

[Mayer, 1992] Richard E. Mayer. Thinking, Problem solving, Cognition. 2nd Edition. New York: Freeman, 1992.

[Mitra and Baral, 2016] Arindam Mitra and Chitta Baral. Learning to use formulas to solve simple arithmetic problems. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 2144–2153, 2016.

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

[Pape, 2004] Stephen J. Pape. 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, 2004.

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

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

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

[Slagle, 1965] James. R. Slagle. Experiments with a deductive question-answering program. J-CACM, 8(12):792-798, 1965.