A Tag-Based Statistical English Math Word Problem Solver with Understanding, Reasoning and Explanation

Chao-Chun Liang, Kuang-Yi Hsu, Chien-Tsung Huang,
Chung-Min Li, Shen-Yu Miao and Keh-Yih Su
Institute of Information Science, Academia Sinica, Taiwan
{ccliang, ianhsu, joecth, cmli, jackymiu, kysu}@iis.sinica.edu.tw

Abstract

This demonstration presents a tag-based statistical English *math word problem* (MWP) solver with understanding, reasoning, and explanation. It analyzes the text and transforms both body and question parts into their tag-based logic forms, and then performs inference on them. The proposed tag (e.g., Agent, Verb, etc.) provides the flexibility for annotating an extracted math quantity with its associated syntactic and semantic information, which can be used to identify the desired operand 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 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 of the MWP cannot be simply extracted by performing keyword/pattern matching. It can clearly show the merit of understanding and inference. (2) 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 procedures 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*.

Previous MWP solvers either adopt rule-based approaches [Mukherjee and Garain, 2008; Hosseini et al., 2014] or purely statistic-based approaches [Kushman et al., 2014; Roy and Roth, 2015] to identify entities, quantities, operations, and get the answer. The main problem of the rule-based approaches is that a wide coverage rule-set is difficult and expensive to construct. It is also awkward in resolving ambiguity problem. In contrast, the main problems of the purely statistic-based approaches is that the performance deteriorates significantly when the MWP is complicated. These approaches are also sensitive to the irrelevant information [Kushman et al., 2014].

A tag-based statistical English MWP solver is thus proposed to perform understanding and reasoning, and avoid the problems mentioned above. The proposed tag-based approach provides the flexibility for annotating an extracted math quantity with its associated syntactic and semantic information, which can be used to identify the desired operand and filter out irrelevant quantities so that the answer can be obtained precisely. 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.

2 Tag-based MWP Solver

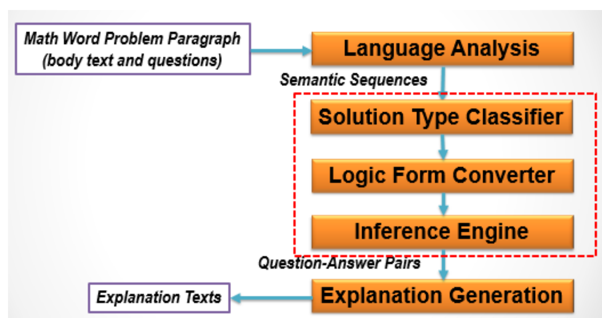

Figure 1: The block diagram of the proposed MWP solver

The block diagram of proposed math word problem solver is shown in Figure 1. First, each sentence in a MWP is analyzed by the *Language Analyzer* module (which adopts Stanford CoreNLP suite [Manning et al., 2014] to generate dependency trees and co-reference chains). The associated linguistic information is then sent to the *Solution Type Classifier* (STC), which is a SVM classifier associated with a linear kernel function, to find out the corresponding math operations. Afterwards, they are converted into the logic form by the *Logic Form Converter* (LFC). The *Inference Engine* (IE) then obtains the answer from those obtained logic expressions. Finally, the *Explanation Generator* module will explain how the answer is obtained according to the given *reasoning chain* [Russel and Norvig, 2009].

2.1 Logic Form Transformation

The LFC non-deterministically generates the domain dependent logic forms (a variant of First-Order-Logic (FOL)) from those crucial math facts associated with quantities and relations between quantities. For example, the sentence "*Fred picks 36 limes.*" will be transformed into the domain dependent logic form as "*quan(q1, #, lime)=36 & verb(q1, pick) & nsubj(q1, Fred)*". Here the associated auxiliary facts "*verb(q1, pick) & nsubj(q1, Fred)*" are our proposed tags to make the system less sensitive to the irrelevant information.

The questions of the MWP will be also transformed into logic functions provided by the IE according to the suggested solution type. Take the question "*How many limes were picked in total?*" as an example. The STC will assign the "*Sum*" operation type to it. Based on that, the LFC will generate the FOL function "*Sum(quan(?q, #, lime), verb(?q, pick)*" to search all quantities that are associated with object "*lime*" and also attached with the verb tag "*pick*".

2.2 Logic Inference

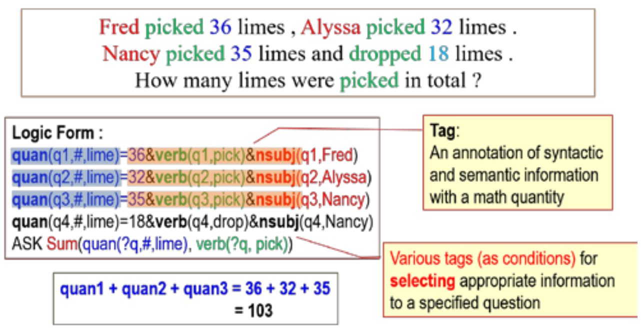

Figure 2: Logic form and logic inference of a *Sum* operation

The IE is used to find out the solution of a MWP. It is responsible for providing utilities to select desired facts and then obtain the answer by taking math operations on selected facts. In addition, it 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, #, lime)*" and with a "*pick*" verb tag, and then perform a *Sum* operation on them. The irrelevant quantity "*quan(q4, #, pear)*" in that example is thus pruned out as its verb tag is "*drop*", not "*pick*". The answer is then obtained by summing those quantities *q1*, *q2* and *q3*.

2.3 Explanation Generation

Based on the reasoning chain generated from the IE (an example is shown in Figure 3), a math operation oriented approach is adopted to explain how the answer is obtained. A specific template is used to generate the explanation text for each kind of operation. For example, the explanation text "*Totally pick 36 limes + 32 limes + 35 limes = 103 limes*" will be generated to explain that the obtained answer is a summation of "*36 limes*", "*32 limes*" and "*35 limes*".

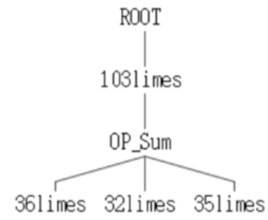

Figure 3: The *Reasoning Chain* from the Inference Engine

3 Demonstration Outline

The MWP solver comprises a web user interface and a processing server. The demonstration will start from an input MWP which will be submitted from the web interface. After that, those modules shown in Figure 1 will be invoked to solve the problem. Once the process is finished, all outputs will be displayed in the web interface: (1) Parse Trees, Dependency and Co-Reference from the language analyzer. (2) Linguistic representations, which are converted from the above language analysis result. (3) Suggested solution type, which identifies the desired math operation that the LFC should adopt. (4) Obtained logical forms, which are transformed from the linguistic representation and specified solution type. (5) Generated reasoning chains and explanation text, which explains how the problem is solved.

4 Conclusion

A tag-based statistical framework is proposed to perform understanding and reasoning for solving English MWPs. The adopted tag can help identify desired operands and filter out irrelevant quantities. Furthermore, by representing the physical meaning of each quantity with tags and using them in the inference process, we can explain how the answer is obtained in a human comprehensible way.

References

[Hosseini et al., 2014] M.J. Hosseini, H. Hajishirzi, O. Etzioni, N. Kushman. Learning to Solve Arithmetic Word Problems with Verb Categorization, *EMNLP*. 2014.

[Kushman et al., 2014] N. Kushman, Y. Artzi, L. Zettlemoyer, R. Barzilay. Learning to Automatically Solve Algebra Word Problems, *ACL*. 2014.

[Manning et al., 2014] C.D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP Natural Language Processing Toolkit, *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*: System Demonstrations, pp. 55-60. 2014.

[Mukherjee and Garain, 2008] A. Mukherjee, U. Garain. A review of methods for automatic understanding of natural language mathematical problems, *Artif Intell Rev*, 2008.

[Roy and Roth, 2015] S.I. Roy and D.I. Roth. Solving General Arithmetic Word Problems, *EMNLP*. 2015.