The 2015 Conference on Computational Linguistics and Speech Processing
ROCLING 2015, pp. 64-70

Explanation Generation for a Math Word Problem Solver

Chien-Tsung Huang (joecth@iis.sinica.edu.tw), Yi-Chung Lin (lyc@iis.sinica.edu.tw) and Keh-Yih Su (kysu@iis.sinica.edu.tw)

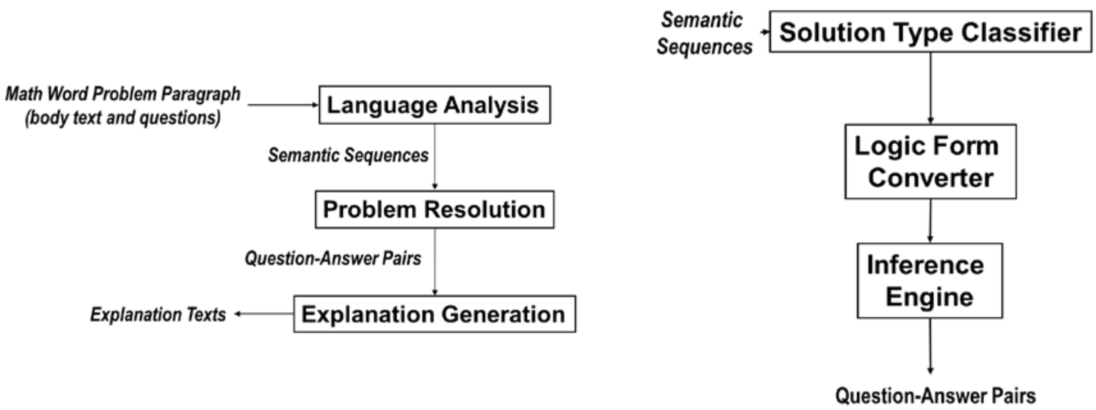
Institute of Information Science[1], Academia Sinica

Extended Abstract:

Background

Machine Reading (MR) aims to make the knowledge contained in the text available in forms that machines can use them for automated processing. That is, machines will learn to read from a few examples and they will read to learn what they need in order to answer questions or perform some reasoning task [1]. Since a domain-independent MR system is difficult to build, the *Math Word Problem* (MWP) [2] is frequently chosen as the first test case to study MR. The main reason for that is that MWP not only has less complicated syntax but also requires less amount of domain knowledge.

The architecture of our proposed approach [3] is shown in Figure 1. First, every sentence in the MWP, including both body text and the question text, is analyzed by the *Language Analysis* module, which transforms each sentence into its corresponding *semantic representation tree*. The sequence of semantic representation trees is then sent to the *Problem Resolution* module, which adopts logic inference approach, to obtain the answer of each question in the MWP. Finally, the *Explanation Generation* (EG) module will explain how the answer is found (in natural language text) according to the given *reasoning chain* [4] (which includes all related logic statements and inference steps to reach the answer).



(a) Math Word Problem Solver Diagram (b) Problem Resolution Diagram

Figure 1. The block diagram of the proposed Math Word Problem Solver.

[1] 128 Academia Road, Section 2, Nankang, Taipei 11529, Taiwan

As depicted in Figure 1(b), the Problem Resolution module in the proposed system consists of three components: *Solution Type Classifier* (TC), *Logic Form Converter* (LFC) and *Inference Engine* (IE). TC is responsible to assign a math operation type for every question of the MWP. In order to perform logic inference, the LFC first extracts the related facts from the given semantic representation tree and then represents them in *First Order Logic* (FOL) *predicates/functions* form [4]. In addition, it is also responsible for transforming every question into an FOL-like utility function according to the assigned solution type. Finally, according to inference rules, the IE derives new facts from the old ones provided by the LFC. Additionally, it is also responsible for providing utilities to perform math operations on related facts.

Besides understanding the given text and then performing inference on it, a very desirable characteristic of a MWP solver (also a MR system) is being able to explain how the answer is obtained in a human comprehensible way. This task is done by the *Explanation Generator* (EG) module, which is responsible to explaining the associated reasoning steps in fluent natural language from the given reasoning chain. In other words, explanation generation is the process of constructing natural language outputs from a non-linguistic input, and is a task of *Natural Language Generation* (NLG).

Various applications of NLG (such as weather report) have been proposed before [5-11]. However, to the best of our knowledge, none of them discusses how to generate the explanation for WMP, which possesses some special characteristics (e.g., math operation[2] oriented description) that are not shared with other tasks. This paper therefore proposes a *math operation oriented approach* to explain how the answer is obtained in solving math word problems.

Proposed Methods

Based on the reasoning chain given by the IE [3], we first search each math operator involved. For each math operator, we generate one sentence. Since explaining math operation does not require complicated syntax, we adopt a specific template to generate the text for each kind of math operator. To the best of our knowledge, this is the first explanation generation that is specifically tailored to the math word problem.

Figure 2 shows the block diagram of our proposed EG. First, the IE generates the answer and its associated reasoning chain for the given math problem. To ease the operation of the EG, we first convert the given reasoning chain into its corresponding *Explanation Tree* (shown at Figure 4) to center around each operator appearing in solving the MWP (which would be convenient to perform sentence segmentation later). Afterwards, the Explanation Tree will be fed into the *Discourse Planner*. The last stage is the *Function Word Insertion & Ordering Module*, which inserts the necessary functional words to the segmented sentences

[2] Where math operations include Sum, Addition, Subtraction, Multiplication, Division, etc.

(resulted from Discourse Planner) and generates the explanation texts according to the selected template (based on the operator encountered).

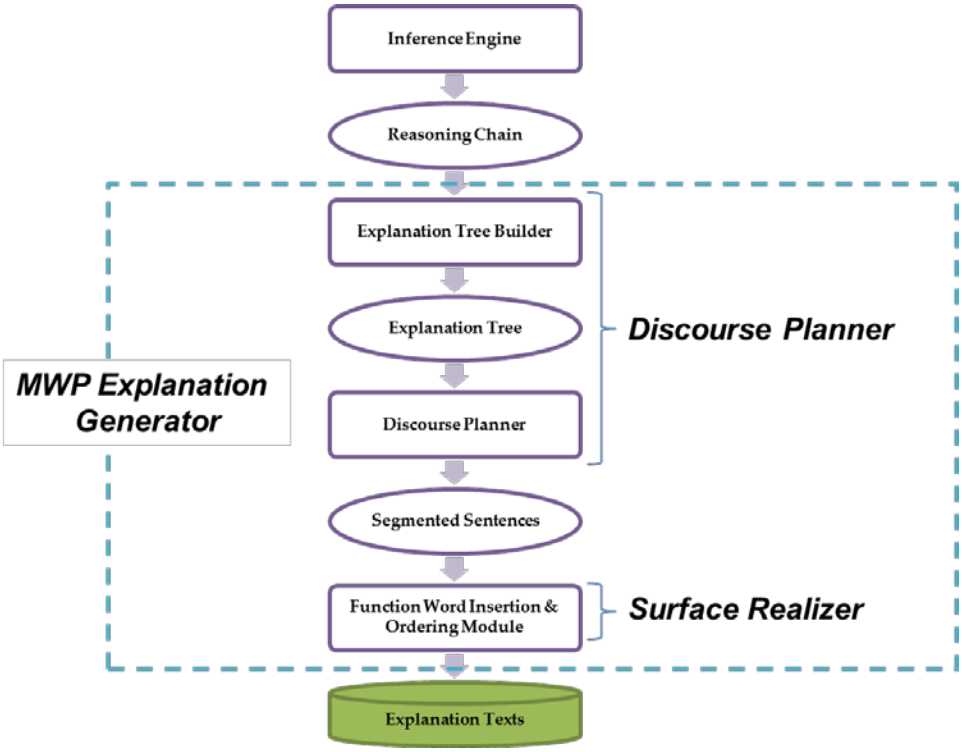


Figure 2. Block Diagram of the proposed MWP Explanation Generator

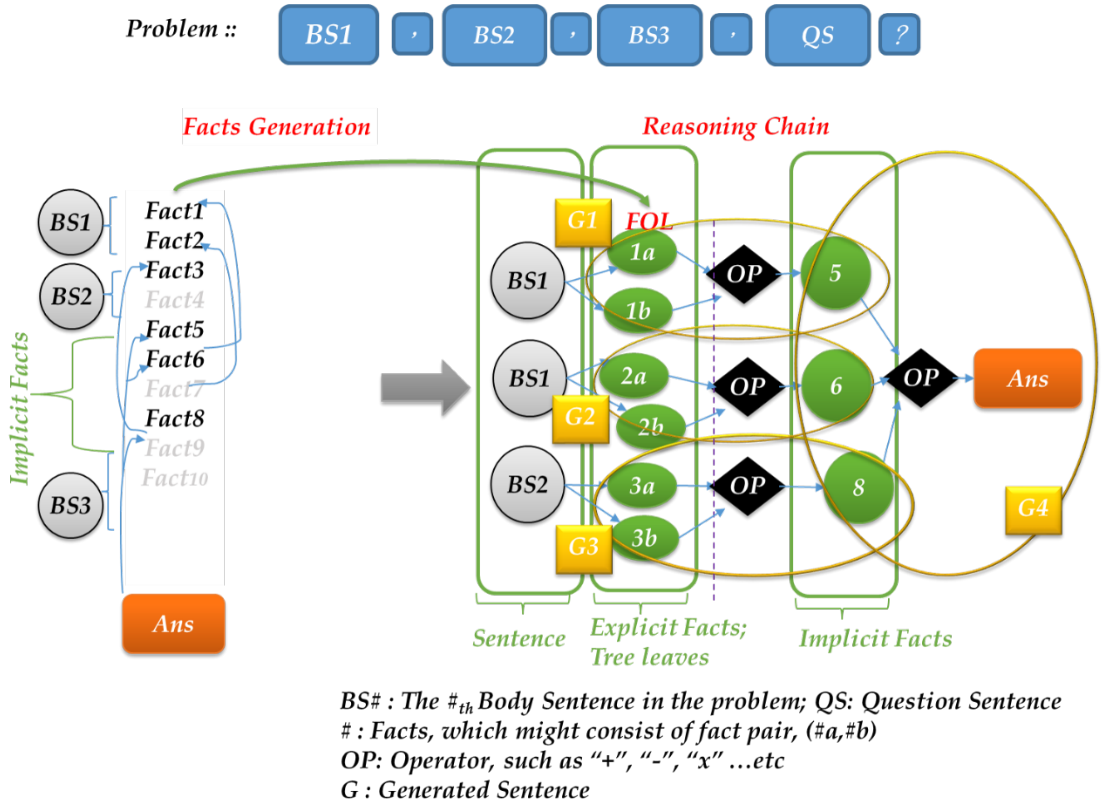
Following example demonstrates how the framework works. And Figure 3(a) reveals more details for each part illustrated in Figure 2.

[Sample-1] 阿志買一臺冰箱和一臺電視機，付 2 疊一萬元鈔票、6 張千元鈔票和 13 張百元鈔票，阿志共付了幾元？

(A-Zhi bought a refrigerator and a TV, paid 2 piles of ten-thousand-dollar bill, six thousand-dollar bill and 13 hundred-dollar bill. How many dollars did A-Zhi totally pay?)

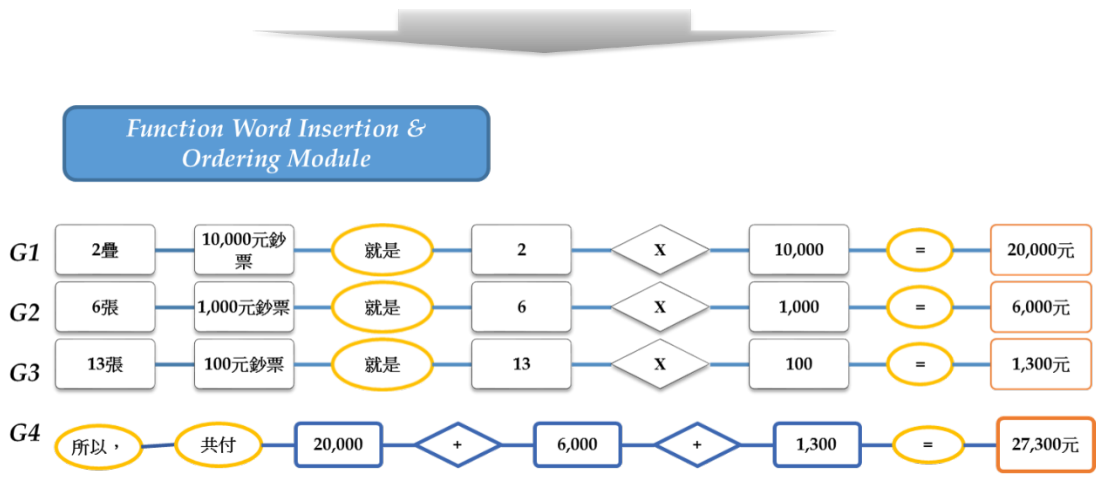
Facts Generation in Figure 3(a) shows how the body text is transformed into meaningful logic facts to perform inference. In math problems, the facts are mostly related to quantities. The generated facts are either the quantities explicitly appearing in the sentence text or the implicit quantities deduced by the IE. Those generated facts are linked together within the reasoning chain constructed by the IE as shown in Figure 3(b). Within this framework, the discourse planner is responsible for selecting the associated content for each sentence to be generated. Figure 3(c) shows how the contents in the Explanation Tree are used as fillers to fill in the template slots for generating the explanation sentences.

A typical reasoning chain, represented with an Explanation Tree structure, is shown at Figure 4. The *operator-node* (OP_node) layers and *quantity-node* (Quan_node) layers are interleaved within the Explanation Tree, and serve as the input to OP Oriented Algorithm in Discourse Planner.



(a) Facts Generation

(b) Reasoning Chain



(c) Function Word Insertion & Ordering Module, serving as the Surface Realizer. It shows how surface realization is done with non-slot fillers (circled by ellipses) and slot-fillers (the diamond shape is for operators, and the rectangle one is for quantities).

Figure 3. (a) Facts Generated from the Body Text. (b) The associated Reasoning Chain, where "G#" shows the facts grouped within the same sentence. (c) Explanation texts

generated by the *Function Word Insertion & Ordering Module* for this example (labeled as G1~G4). Except those ellipses which symbolize non-slot fillers, other shapes denote slot-fillers. Furthermore, Diamond symbolizes OP_node while Rectangle symbolizes Quan_node.

Also, as shown at Figure 3(b), the (*#a*, *#b*) pair denotes facts derived from the body sentences. The *OP* means the operator used to deduce implicit facts and represented as non-leaf circle nodes. Each "*G?*" expresses a sentence to be generated. Given the reasoning chain, the first step is to decide how many sentences will be generated, which corresponds to the *Discourse Planning* phase [12] of the traditional NLG task. Currently, we will generate one sentence for each operator shown in the reasoning chain. For the above example, since there are four operators (three IE-Multiplication[3] and one LFC-Sum in Figure 4), we will have four corresponding sentences; and the associated nodes (i.e., content) are circled by "*G?*" for each sentence in the figure.

Furthermore, Figure 4 shows that three sets of facts are originated from the 2nd body sentence (indicated by three *S2* nodes). Each set contains a corresponding quantity-fact (e.g., *q1*(疊), *q2*(元), and *q3*(張)) and its associated object (e.g., *n1*, *n2*, and *n3*). For example, the first set (the left most one) contains *q1*(疊) (for "2 疊") and *n1* (for "一萬元鈔票"). This figure also shows that the outputs of three IE-Multiplication operators (i.e., "20,000 元", "6,000 元", and "1,300 元") will be fed into the last LFC-Sum to get the final desired result "27,300 元" (denoted by the "Ans(SUM)" node in the figure).

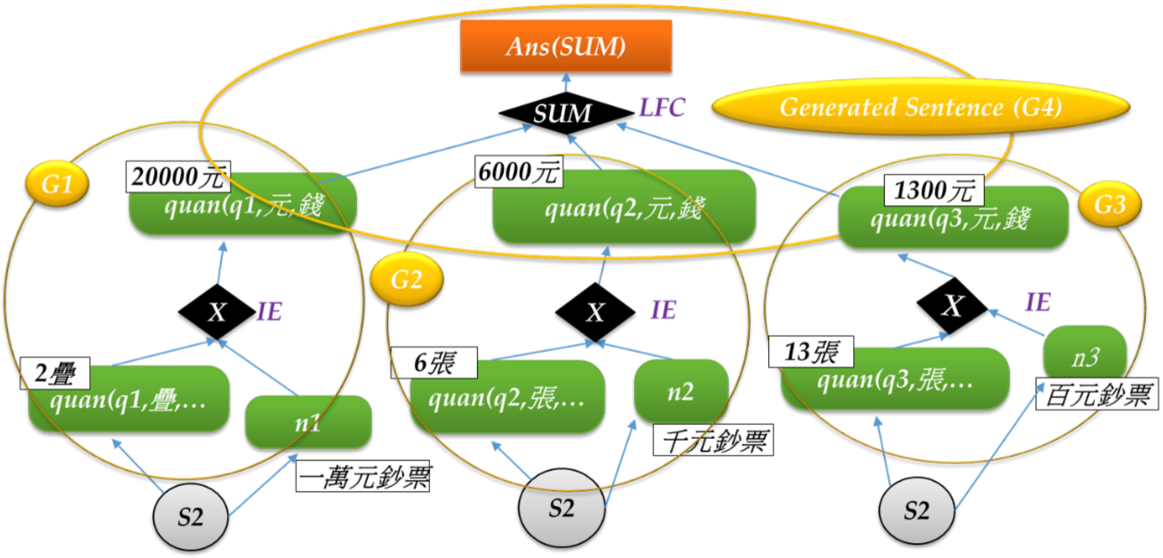


Figure 4. Explanation Tree for Discourse Planning, where *S2* means the facts from the 2nd body sentence.

[3] Prefixes "IE-" and "LFC-" denote that those operators is issued by IE and LFC, respectively.

Our EG of the MWP solver is able to explain how the answer is resulted in a human comprehensible way, where the related reasoning steps can be systemically accomplished from the giving reasoning chain according to the specified template.

The main contributions of this paper are shown as follows,

1. *The Explanation Tree is introduced for facilitating the discourse planning on MWP.*
2. *An operator oriented algorithm is proposed to segment the Explanation Tree into various sentences, which makes our Discourse Planner universal for math word problems regardless of the language adopted.*
3. *We propose using operator-based templates to generate the natural language text for explaining the associated math operation.*

Admittedly, the work related to multi-template per operator can be further explored after examining more cases. In this case, a statistical model would be required to select the most appropriate template for each given operation..

References

[1] S. Strassel, D. Adams, H. Goldberg, J. Herr, R. Keesing, D. Oblinger, H. Simpson, R. Schrag, J. Wright, *The DARPA Machine Reading Program - Encouraging Linguistic and Reasoning Research with a Series of Reading Tasks*, LREC, (2010).

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

[3] Y.C. Lin, C.C. Liang, K.Y. Hsu, C.T. Huang, S.Y. Miao, W.Y. Ma, L.W. Ku, C.J. Liau, K.Y. Su, *Designing a Tag-Based Statistical Math Word Problem Solver with Reasoning and Explanation*, to be published at International Journal of Computational Linguistics and Chinese Language Processing (IJCLCLP), (2016).

[4] S.J. Russell, *Artificial Intelligence : A Modern Approach (3e)* (2009).

[5] M.A.K. Halliday, *An Introduction to Functional Grammar.*, Edward Arnold, London, (1985b).

[6] E. Goldberg, N. Driedger, R. Kittredge, *Using natural-language processing to produce weather forecasts*, IEEE Expert, 9 (1994) 45-53.

[7] C. Paris, K. Vander Linden, *Drafter: An interactive support tool for writing multilingual instructions*, IEEE Computer, 29 (1996) 49-56.

[8] M. Milosavljevic, *Content selection in comparison generation*, In Proceedings of the 6th European Workshop on Natural Language Generation, Duisburg, Germany, (1997 March) 72-81.

[9] C. Paris, K. Vander Linden, S. Lu, *Automatic document creation from software specifications*, Proceedings of the 3rd Australian Document Computing Symposium (ADCS-98), (1998) 26-31.

[10] J. Coch, *Interactive generation and knowledge administration in MultiM´et´eo*, In

Proceedings of the Ninth International Workshop on Natural Language Generation, (1998 Aug).

[11] E. Reiter, R. Robertson, L. Osman, *Types of knowledge required to personalise smoking cessation letters,* In Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making. Springer-Verlag, (1999).

[12] D. Jurafsky, J.H. Martin, *Speech and Language Processing*, Chapter 20, Prentice Hall, Englewood Cliffs, New Jersey (2000).