The 2015 Conference on Computational Linguistics and Speech Processing
ROCLING 2015, pp. 58-63

Designing a Tag-Based Statistical Math Word Problem Solver with Reasoning and Explanation

Yi-Chung Lin (lyc), Chao-Chun Liang (ccliang), Kuang-Yi Hsu (ianhsu), Chien-Tsung Huang (joecth), Shen-Yun Miao (jackymiu), Wei-Yun Ma (ma), Lun-Wei Ku (lwku), Churn-Jung Liau (liaucj), and Keh-Yih Su (kysu@iis.sinica.edu.tw)

Institute of Information Science[1], Academia Sinica

Extended Abstract:

Background

Since *Big Data* mainly aims to explore the correlation between surface features but not their underlying causality relationship, the *Big Mechanism*[2] program has been proposed by DARPA to find out "why" behind the "Big Data". However, the pre-requisite for it is that the machine can read each document and learn its associated knowledge, which is the task of *Machine Reading* (MR). Since a domain-independent MR system is complicated and difficult to build, the math word problem (MWP) [1] is frequently chosen as the first test case to study MR (as it usually uses less complicated syntax and requires less amount of domain knowledge).

According to the framework for making the decision while there are several candidates, previous MWP algebra solvers can be classified into: (1) Rule-based approaches with logic inference [2-7], which apply rules to get the answer (via identifying entities, quantities, operations, etc.) with a logic inference engine. (2) Rule-based approaches without logic inference [8-13], which apply rules to get the answer without a logic inference engine. (3) Statistics-based approaches [14, 15], which use statistical models to identify entities, quantities, operations, and get the answer. To our knowledge, all the statistics-based approaches do not adopt logic inference.

The main problem of the rule-based approaches mentioned above is that the coverage rate problem is serious, as rules with wide coverage are difficult and expensive to construct. Also, since they adopt Go/No-Go approach (unlike statistical approaches which can adopt a large Top-N to have high including rates), the error accumulation problem would be severe. On the other hand, the main problem of those approaches without adopting logic inference is that they usually need to implement a new handling procedure for each new type of problems (as the general logic inference mechanism is not adopted). Also, as there is no inference engine to generate the *reasoning chain* [16], additional effort would be required for

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

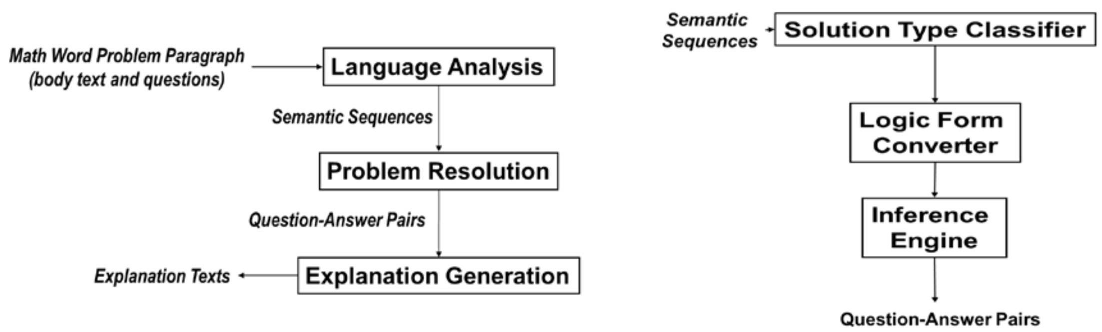
[2] http://www.darpa.mil/Our_Work/I2O/Programs/Big_Mechanism.aspx

generating the explanation.

To avoid the problems mentioned above, a *tag-based statistical framework* which is able to perform understanding and *reasoning with logic inference* is proposed in this paper. It analyzes the body and question texts into their associated tag-based[3] logic forms, and then performs inference on them. Comparing to those rule-based approaches, the proposed statistical approach alleviates the ambiguity resolution problem, and the tag-based approach also provides the flexibility of handling various kinds of possible questions with the same body logic form. On the other hand, comparing to those approaches not adopting logic inference, the proposed approach is more robust to the irrelevant information and could more accurately provide the answer. Furthermore, with the given reasoning chain, the explanation could be more easily generated.

Proposed Framework

The main contributions of our work are: (1) proposing a tag-based logic representation such that the system is more robust to the irrelevant information and could provide the answer more precisely; (2) proposing a unified statistical framework for performing reasoning from the given text.



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

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

The block diagram of the proposed MWP solver 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 the logic inference approach to obtain the answer for each question. Finally, the *Explanation Generation* (EG) module will explain how the answer is

[3] The associated *modifiers* in the logic form (such as verb(q1,進貨), agent(q1,文具店), head($n1_p$,筆), color($n1_p$,紅), color($n2_p$,藍) in the example of the next page) are regarded as various *tags* (or conditions) for selecting the appropriate information related to the question specified later.

obtained (in natural language text) according to the given reasoning chain.

As the figure depicted, the Problem Resolution module in our system consists of three components: *Solution Type Classifier* (TC), *Logic Form Converter* (LFC) and *Inference Engine* (IE). TC suggests a way to solve the problem for every question in an MWP. In order to perform logic inference, the LFC first extracts the related facts from the given semantic representation tree and then represents them as *First Order Logic* (FOL) *predicates/functions* [16]. It also transforms each 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. Besides, it is also responsible for providing utilities to perform math operations on related facts.

Take the MWP "文具店進貨2361枝紅筆和1587枝藍筆 (A stationer bought 2361 red pens and 1587 blue pens), 文具店共進貨幾枝筆 (How many pens did the stationer buy)?" as an example. Figure 2 shows the *Semantic Representation* of this example.

```
{進貨.buy|買:
    agent={文具店},
    theme={和.and(
        {筆.PenInk|筆墨:
            quantity={2361},
            color={紅.red|紅}
        },
        {筆.PenInk|筆墨:
            quantity={1587},
            color={藍.blue|藍}
        }
    )},
}
```

Figure 2 (a)

```
{進貨.buy|買:
    agent={文具店},
    共.quantity={all|全},
    theme={筆.PenInk|筆墨:
        幾.quantity={Ques|疑問}
    },
}
```

Figure 2 (b)

Figure 2. Semantic Representation of (a)"文具店進貨 2361 枝紅筆和 1587 枝藍筆 (A stationer bought 2361 red pens and 1587 blue pens), (b)文具店共進貨幾枝筆 (How many pens did the stationer buy)?"

Based on the semantic representation given above, the TC will assign the operation type "Sum" to it. The LFC will then extract the following two facts from the first sentence:

quan(q1,枝,$n1_p$)=2361&verb(q1,進貨)&agent(q1,文具店)&head($n1_p$,筆)&color($n1_p$,紅)

quan(q2,枝,$n2_p$)=1587&verb(q2,進貨)&agent(q2,文具店)&head($n2_p$,筆)&color($n2_p$,藍)

The quantity-fact "2361 枝紅筆 (2361 red pens)" is represented by "quan(q1,枝,$n1_p$)=2361", where the argument "$n1_p$"[4] denotes "紅筆 (red pens)" due to the facts "head($n1_p$,筆)" and "color($n1_p$,紅)". Likewise, the quantity-fact "1587 枝藍筆 (1587 blue pens)" is represented by "quan(q2,枝,$n2_p$)=1587". The LFC also issues the utility call "ASK Sum(quan(?q,枝,筆),verb(?q,進貨)&agent(?q,文具店))" (based on the assigned solution type) for the question. Finally, the IE will select out two quantity-facts "quan(q1,枝,$n1_p$)=2361" and "quan(q2,枝,$n2_p$)=1587", and then perform "Sum" operation on them to obtain "3948".

If the question in the above example is "文具店共進貨幾枝紅筆 (How many red pens did the stationer buy)?", the LFC will generate the following facts and utility call for this new question:

head($n3_p$,筆)&color($n3_p$,紅)

ASK Sum(quan(?q,枝,$n3_p$),verb(?q,進貨)&agent(?q,文具店))

As the result, the IE will only select the quantity-fact "quan(q1,枝,$n1_p$)=2361", because the modifier in QLF (i.e., "color($n3_p$,紅)") cannot match the associated modifier "藍 (blue)" (i.e., "color($n2_p$,藍)") of "quan(q2,枝,$n2_p$)=1587". After performing "Sum" operation on it, we thus obtain the answer "2361". (We will skip EG due to space limitation. Please refer to [17] for the details).

Preliminary Results

Currently, we have completed all the associated modules (including Word Segmenter, Syntactic Parser, Semantic Composer, TC, LFC, IE, and EG), and have manually annotated 75 samples (in our elementary school math corpus) as the seed corpus (with syntactic tree, semantic tree, logic form, and reasoning chain annotated). Besides, we have cleaned the original elementary school math corpus and encoded it into the appropriate XML format. There are total 23,493 problems divided into six grades; and the average number of words of the body text is 18.2 per problem. Table 3 shows the statistics of the converted corpus.

We have completed a prototype system and have tested it on the seed corpus. The success of our pilot run has demonstrated the feasibility of the proposed approach. We plan to use the next few months to perform *weakly supervised learning* [18] and fine tune the system.

[4] The subscript "p" in "$n1_p$" indicates that "$n1_p$" is a *pseudo* nonterminal derived from the nonterminal "n1", which has four terminals "2361", "枝", "紅" and "筆". More details about pseudo nonterminal will be given at Section 2.3.

Table 1. MWP corpus statistics and Average length per problem

Corpus	Num. of problems
Training Set	20,093
Develop Set	1,700
Test Set	1,700
Total	23,493

MWP corpus statistics

Corpus	Avg. Chinese Chars.	Avg. Chinese Words
Body	27	18.2
Question	9.4	6.8

Average length per problem

References

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

[2] D.G. Bobrow, *Natural language input for a computer problem solving system*, Ph.D. Dissertation, Massachusetts Institute of Technology, (1964).

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

[4] E. Charniak, *CARPS, a program which solves calculus word problems*, Report MAC-TR-51, Project MAC, MIT, (1968).

[5] E. Charniak, *Computer solution of calculus word problems*, In Proc. of International Joint Conference on Artificial Intelligence, (1969).

[6] D. Dellarosa, *A computer simulation of children's arithmetic word-problem solving*, Behavior Research Methods, Instraments, & Computers, 18 (1986) 147-154.

[7] Y. Bakman, *Robust Understanding of Word Problems With Extraneous Information*, (2007 Jan).

[8] J.P. Gelb, *Experiments with a natural language problem solving system*, In Pros. of IJCAI-71, (1971).

[9] B. Ballard, A. Biermann, *PROGRAMMING IN NATURAL LANGUAGE : "NLC" AS A PROTOTYPE*, ACM-Webinar, (1979).

[10] A.W. Biermann, B.W. Ballard, *Toward Natural Language Computation* American Journal of Computational Linguistic, 6 (1980 April-June).

[11] A. Biermann, R. Rodman, B. Ballard, T. Betancourt, G. Bilbro, H. Deas, L. Fineman, P. Fink, K. Gilbert, D. Gregory, F. Heidlage, *INTERACTIVE NATURAL LANGUAGE PROBLEM SOLVING:A PRAGMATIC APPROACH* In Pros. of the first conference on applied natural language processing, (1982).

[12] C.R. Fletcher, *COMPUTER SIMULATION - Understanding and solving arithmetic word problems: A computer simulation*, Behavior Research Methods, Instruments, & Computers, 17 (1985) 565-571.

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

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

[15] S.I. Roy, T.J.H. Vieira, D.I. Roth, *Reasoning about Quantities in Natural Language*, TACL, 3 (2015) 1-13.

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

[17] C.T. Huang, Y.C. Lin, K.Y. Su, *Explanation Generation for a Math Word Problem Solver*, to be published at International Journal of Computational Linguistics and Chinese Language Processing (IJCLCLP), (2016).

[18] Y. Artzi, L. Zettlemoyer, *Weakly supervised learning of semantic parsers for mapping instructions to actions*, Transactions of the Association for Computational Linguistics, 1 (2013) 49-62.