International Journal of Scientific Research and Engineering Development-– Volume 3 Issue 3, May – June 2020

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Arithmetic Word Problem Solver using Unit Dependency Graph and Verb Categorization

Khin Cho Win*, Naw Lay Wah**

*(Faculty of Information Science, *University of Computer Studies (Mandalay)*, and Myanmar
khinchowin@ucsm.edu.mm)

** (Faculty of Information Science, *University of Computer Studies (Mandalay)*, and Myanmar
nawlaywah@ucsm.edu.mm)

--************************----------------------------------

Abstract:

Nowadays, the arithmetic questions that are expressed in natural language such as English are hugely getting interest by researchers. Although some useful researches have been proposed to solve word problems, there are still gaps in implementing a robust arithmetic word problem solver as the answers of the word problems cannot be easily extracted with the approach of keyword or pattern matching. According to this motivation, this research focuses on generating the correct equation from the word problem and deriving the solution. The aim of this proposed work is to implement an arithmetic word problem solver that can understand the elementary math word problems, derive the symbolic equation, and generate the result from the equation. The system is implemented with the combination of the verb semantics and the graph. The elementary student can obtain many benefits since the system is resulted the equation along with the answers.

Keywords **—Mathematical Word Problem (MWP), Natural Language, Arithmetic Word Problem, Word Problem Solver.**

--************************----------------------------------

I. INTRODUCTION

Natural language processing (NLP) is primarily concerned with the computer interaction and artificial intelligence (AI) that make computers to reason and inferenceautomatically. Since Bobrow [1] first implemented the STUDENT program for his dissertation project, there are many word problem solvers that are conducted by NLP and AI researchers. Unfortunately, the answers to the word problems cannot be easily obtained with the approach of keyword/ pattern matching.

In an arithmetic word problem, a partial world state is described first, some information of the state is then updated or elaborated and finally a quantitative entity is asked. The normal children can easily learn the problem-solving skills such as the understanding of large vocabulary, the utilization of world knowledge, the proficiency of the syntax structure, and the capability of integrating individual sentences into a reasonable model. On the other hand, making sense of the story problem which explains arbitrary activities such as buying clothes, going to the games, and picking seashells is very challenging for the machines. Due to the aforementioned difficulties, solving the mathematic word problems has been becoming the challenging research work. And we need to consider which type of the problems are solved and collect the dataset carefully.

The university-level research papers have been experimenting to handle the integrals problem since 2013. According to the Todai Robot Project [2], the integrals at the university entrance exam is easier than the elementary exam questions. The system needs to correctly understand the concept and the relationshipsbetween concepts. In the example word problem "June has 18 journals and 24 magazines in the library. He bought 19 journals at a book shop yesterday. How many journals does he have now?", we need to extract not only the concepts "journals", "magazines", "library" and "book shop" but also the relationship between "library" and "book shop". Without extracting those, it is not possible to derive a correct equation from the problem.

The arithmetic word problem solver is needed for the user to input the correct elementary word problem. The word problem solver is implemented with the graph and the verb semantics. The graph checks the unit relations and the verb semantics is utilized to determine the operation between the units. The detail will be explained in the corresponding section.

The rest of this paper is organized as follows. The next section reviews the related work and system architecture, the proposed algorithm, evaluation and discussion, and the final conclusion will be described in this order.

II. RELATED WORK

Roy, Vieira and Roth [3] introduced a quantity reasoning system at the elementary-level mathematical word problems. Their limitation is that the word problems must contain only two or three quantities. The co-reference resolution and semantic analysis is needed to consider.

ARIS[4]proposed the elementary-level arithmetic solver with the entity-container approach. This approach was semantically updated the states of each entity by the categorization of the contained verbs in the problem statement. ARIS required extra overheads so that it annotated each verb with the corresponding verb class. Another drawback isthat the verb semantics was mainly focused to solved addition and subtraction arithmetic word problems. Moreover, the system did not focus to solve the set completion problem type since there was missing information in it. Most of the errors were due to the irrelevant information, parsing and co-reference resolution.

Roy and Roth [5] proposed to solve the elementary math questions on the web portal. They exploited a context-free grammar (CFG) parser and an arithmetic problem solver [6]. The limitation is that it can only simple elementary word problems.

Win and Wah [7] presented an automatic answering system to solve arithmetic word problems. They utilized the number quantifier [3] and the unit dependency graph [8]. The system cannot extract the arithmetic operation (addition, subtraction, multiplication and division). Another problem is that the compare problem cannot be solve correctly.

The current paper proposed to take advantage of the graph and the verb semantics [4].

III. SYSTEM ARCHITECTURE

This section is to correctly predict the relevant quantity with the graph and the corresponding arithmetic operation (addition, subtraction, multiplication and division) with the combination approach of verb semantics and the graph.

A. Verb Semantics

Semantics study the branch of linguistics and logic that is specialized in meaning. Many words have the similar meaning but it is very crucial that subtle differences are identified between them. For example, 'anger' and 'rage' are identical in meaning (synonyms) but 'rage'indicates a greater human reaction to a situation than 'hate'.

The word verb derives from the Latin word 'verbum'. In the language, the verb of a sentenceplays an important role in deciding whether it describes the state and information or it conveys the subject into motion. Classifying the verb is a very interesting subject of computational linguistics research since understanding the role of verbs is important in conveying natural language semantics. In addition, generalization based on verb classification is of great important to many natural language applications, ranging from shallow semantic parsing to semantic search and information extraction.

To automatically understand the meaning of words in the word problem, the verb categories are needed to train for each sentence. Training the verb categories is done by SVM. In computational linguistics, training the verb categories is necessary to predict the label (increasing, decreasing, or unchanging) for each (unit, verb) pair in the sentence. The verb classification concept is similar to ARIS [4]. The seven categories of verb are shown in Figure 1.

Verb Category	Example Sentence
Observation	There were 28 bales of hay in the barn.
Positive	Joan went to 4 football games this year.
Negative	John lost 3 of the violent balloons.
Positive Transfer	Mike's dad borrowed 7 nickels form Mike.
Negative Transfer	Jason placed 131 erasers in the drawer.
Construct	Karen added 0.25 of a cup of walnuts to a batch of a trail mix.
Destroy	The rabbits ate 4 of Dan's potatoes.

Fig. 1Seven verb categories and the corresponding example

Out of seven verb categories, 3 categories are concerned for one argument and the remaining 4 categories are related to two arguments. The first 3 categories are listed as follows:

1) ***Observation***: The quantity from the problem statement is initialized in the argument. Generally, this verb categories are found in the first sentence of the MWP.
2) ***Positive***: The argument is updated the action and the quantity is increased in this argument.
3) ***Negative***: The argument is updated the action and the quantity is decreased in this argument.

The remaining four verb categories that are concerned with two arguments are

1) ***Positive Transfer***: The second argument transfers the quantity into the first argument.
2) ***Negative Transfer***: The first argument transfers the quantity into the second argument.
3) ***Construct***: The quantities of both arguments are increased.
4) ***Destroy***:The quantities of both arguments are decreased.

B. Unit Dependency Graph

The main purpose of the unit dependency graph (UDG) is to capture the dependency between units of two quantities and the operation dependency, especially multiplication and division. The graph needs two classifier: a binary classifier for the quantity and a multiclass classifier for the operation. We use structured support vector machine (SSVM) for the classifier. The lowest common ancestor (LCA) of two quantities (q_i, q_j) is predicted for theoperation. A monotonic expression tree, a binary tree representation of a mathematical expression, is utilized to generate the correct equation. In the expression tree, the leaf is the number and the internal node is the operation between two numbers.

For an expression E, let T be an expression tree for E , $S(E)$ be the set of all quantities in the problem that are not used in E, q be the quantity, λ be the operation threshold, λ_{IRR} be the quantity threshold . The score is computed as follows:

$$Score(E) = \sum_{q\in E\ \wedge LABEL(q)=RATE} P(q) + \lambda \sum_{e\in E} P(e) + \lambda_{IRR} \sum_{q\in S(E)} IRR(q) + \sum_{q_i,q_j} LCA(q_i, q_j, \pi) \quad (1)$$

Where,

$IRR(q)$ = irrelevant quantity classifier for the quantity q

π = the operation between two quantities q_i and q_j.

Currently, we assume that the valid tree expressions ($TREES$) are achieved from the underlying mathematical word problem. Among many tree expressions, we apply the inference algorithm as in (2).

$$\arg \max_{T\in TREES} Score(E) \quad (2)$$

The graph approach is exploited to determine not only whether two units are the same but also there exists rate unit. The rate unit is defined as the one that one quantity q_i has two units and one of the units is the unit of other quantity q_j. The rate unit can be classified into explicit rate (e.g., 50 miles per hour) and implicit rate (e.g., each student has 3 books.). As the graph takes the form of “Numerator per Denominator”, we need to extract numerator and denominator.

UDG exploits two classifiers: vertex classifier and edge classifier. The vertex classifier is a binary classifier that determines whether the given vertex of UDG is rate. The labels of the UDG vertex classifier are “Rate” and “Not Rate”. The considered features for that classifier are context features (unigrams, bigrams, POS tags and their conjunctions from the vertex neighborhood) and rule-based features.

Problem
At the town carnival Billy rode the ferris wheel 7.0 times and the bumper cars 3.0 times. If each ride cost 5.0 tickets, how many tickets did he use?
Solution: (5.0*(7.0+3.0))
Expression Tree

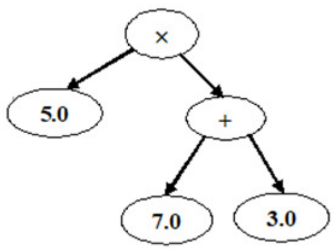

Fig. 2 A word problem, its solution along with the binary expression tree

TABLE I
UNITS OF UNIT DEPENDENCY GRAPH

	Text Mention	**Numerator**	**Denominator**
Explicit Rate	50 miles per hour	miles	hours
Implicit Rate	each student has 3 books.	books	student

Learning Edge Label of UDG

Input: Monotonic expression tree T, vertex pairs v_i, v_j, and their corresponding vertex labels
Output: Label of edge between v_i and v_j
$path \leftarrow Path(T, v_i, v_j)$
$CountMulDiv$
$\leftarrow Number\ of\ Multiplication\ and\ Division$
$nodes\ in\ path$
if v_i and v_j have same vertex label, and $countMulDiv = 0$ then
 return SAME UNIT
end if
if v_i and v_j have different vertex labels, and $countMulDiv = 1$ then
 if path contains × and v_i is RATE then
 return $RATE_{\overrightarrow{DENOMINATOR}}$
 end if
 if path contains × and v_j is RATE then
 return $RATE_{\overleftarrow{DENOMINATOR}}$
 end if
 if path contains ÷ and v_i is RATE then
 return $RATE_{\overrightarrow{NUMERATOR}}$
 end if
 if path contains $\div_r$ and v_j is RATE then
 return $RATE_{\overleftarrow{NUMERATOR}}$
 end if
end if
return Cannot determine edge label

Fig. 3 Edge label classifier of UDG

The edge classifier is a multi-class classifier that determines the edge relation between a pair of nodes of the UDG. The six labels are “same unit”, “numerator unit of the source vertex”, “denominator unit of the source vertex”, “numerator unit of the destination vertex”, “denominator unit of the destination vertex” and “no relation”. The edge classifier’s features are context-based features and rule-based features. The edge classifier algorithm is illustrate in Fig. 3.

IV. PROPOSED ALGORITHM

Input: a valid mathematical word problem

Output: an equation and its solution

Begin

 Preprocess the input problem

 Simplify the problem

 Determine that the number is relevant

 Determine that the unit of the quantity is rate

 for each of two relevant quantities

  Determine the operation

  Derive the symbolic equation

 Return the equation and its solution

End

V. EVALUATION AND DISCUSSION

The system is implemented on the platform of window 10 x64-based processor with the installed memory of 12.0 GB. The processor is Intel ® Core ™ i5-8250U CPU @ 1.60 GHz 1.80 GHz.

A. Dataset

The system is evaluated the performance on the AllArith dataset from the UDG [8]. The total number of mathematical word problems is 831 problems. The considered problem types are the multi-step arithmetic word problems that are composed of addition, subtraction, multiplication and division.

B. Experiment and Discussion

The performance of the system is evaluated with the benchmark dataset AllArith. Correctly extracting relevant number quantity is illustrated in table II and table III.

TABLE III
TOTAL NUMBER OF CLASSIFYING QUANTITIES

	Identify as Relevance	Identify as Irrelevance	Total
Relevance	381	3	384
Irrelevance	10	15	25
Total	391	18	409

TABLE IIIII
PERFORMANCE MEASURE OF NUMBER CLASSIFIER

Precision	97%
Recall	99%
F-Measure	98%
Precision	97%

The proposed work is compared with the graph and the verb semantic approach. The performance comparison is illustrated in table IV and table V.

TABLE IVV
PERFORMANCE MEASURE OF RELEVANT QUANTITIES

	AllArith DataSet
Verb Semantic Approach	82%
Graph Based Approach	90%
Proposed Approach	93%

TABLE V
PERFORMANCE MEASURE OF DERIVING CORRECT SYMBOLIC EQUATION

	AllArith DataSet
Verb Semantic Approach	29%
Graph Based Approach	72%
Proposed Approach	75%

The major issue is the parsing of subject from the input problem such as "There are 11.0 rulers and 34.0 crayons in the drawer . Tim placed 14.0 rulers in the drawer . How many rulers are now there in all ?" Another challenge is that the current research work does not considered the verb such as crack, destroy and broken. Another issues is the misunderstanding of the relevant number detection such as "At the hardware store , 0.25 the nails are size 2.0d and 0.5 the nails are size 4.0d . What fraction of the nails are either size 2.0d or 4.0d ?"

VI. CONCLUSIONS

Although many researchers have been proposed to solve arithmetic word problems, there are still limitations and gaps in developing a robust arithmetic word problem solver. Therefore, the arithmetic word problem solver that can correctly derive the equation of the symbols is proposed. The current system is more robust to the only graph-based approach and the only verb-semantics approach. Since the system can solve all four basic operations and the main sources of the data are from math- aids.com and ixl.com that are focused for third, fourth and fifth grade students, those graders can benefit from this system. In the near future, the system will be advanced to improve the performance.

REFERENCES

[1] D. G. Bobrow, *Natural Language Input for a Computer Problem Solving System*, Sep. 1964, Accessed: Apr. 27, 2020. [Online]. Available: https://dspace.mit.edu/handle/1721.1/6903.

[2] H. Arai and T. Matsuzaki, "The impact of A.I. on education – Can a robot get into The University of Tokyo?," p. 9.

[3] S. Roy, T. Vieira, and D. Roth, "Reasoning about Quantities in Natural Language," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 1–13, Dec. 2015, doi: 10.1162/tacl_a_00118.

[4] M. J. Hosseini, H. Hajishirzi, O. Etzioni, and N. Kushman, "Learning to Solve Arithmetic Word Problems with Verb Categorization," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 523–533, doi: 10.3115/v1/D14-1058.

[5] S. Roy and D. Roth, "Illinois Math Solver: Math Reasoning on the Web," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, San Diego, California, 2016, pp. 52–56, doi: 10.18653/v1/N16-3011.

[6] S. Roy and D. Roth, "Solving General Arithmetic Word Problems," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015, pp. 1743–1752, doi: 10.18653/v1/D15-1202.

[7] K. C. Win and N. L. Wah, "Development of an Automatic Answering System to Arithmetic Questions," in *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, Jun. 2018, pp. 281–285, doi: 10.1109/ICIS.2018.8466459.

[8] S. Roy and D. Roth, "Unit Dependency Graph and its Application to Arithmetic Word Problem Solving," *arXiv:1612.00969 [cs]*, Dec. 2016, Accessed: Mar. 05, 2020. [Online]. Available: http://arxiv.org/abs/1612.00969.