WALGES: Weighted Probability Based Scoring Approach for Solving Algebraic Word Problems using Semantic Parsing

Habibur Rahman
Department of Computer Science & Engineering
University of Information Technology & Sciences
Dhaka, Bangladesh
habib.rahman@uits.edu.bd

Julia Rahman
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology
Rajshahi, Bangladesh
juliacse06@gmail.com

Asmaul Husna
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology
Rajshahi, Bangladesh
tithi_cse_ruet@yahoo.com

***Abstract*—Virtual assistants like Google Assistant, Siri, Cortana, and many others, are now the major feature of smartphones and tablets. Natural language understanding is the most promising part of those. Besides question and answering, sometimes those smart assistants face verbally stated mathematical problems. Algebraic word problems are one of the fundamental mathematical problems where financial news, election results, sports results, and many other services are also related. In this paper, we have generated solutions of the algebraic word problems. Semantic parsing is used to ground the problem text into containers, quantities, and entities to generate equation trees. From the equation trees, features are extracted to train a local and a global classifier model. After that, to find the candidate equation from the generated equation, probabilistic scores from the local and the global classifier models have used. Backpropagation neural network algorithm is used to incorporate the probabilistic scores of the local and the global classifier models with weights. Our proposed weighted probability based scoring approach increases the accuracy to predict the solution of the algebraic word problems about 9% surpassing the existing system. Our result showed that our proposed technique is useful.**

***Index Terms*—Natural Language Understanding, Natural Language Processing, Semantic Parsing and Reasoning, Information Engineering**

I. INTRODUCTION

Virtual assistants are the software that can understand and respond to instructions in natural language. They can also be called smart assistants or bots. To make virtual assistants more effective, it needs to solve mathematical problems that are described in text or voice. Verbally expressed addition, subtraction, multiplication, and division problems are the algebraic word problem. An example of the algebraic word problem is shown in the Fig. 1:

Melanie had 19 dimes in her bank. Her dad gave her 39 dimes and her mother gave her 25 dimes. How many dimes does Melanie have now?

Fig. 1. Example of Algebraic Word Problem (Addition)

Math word problem solving can be divided into two different categories. One of them is the symbolic approach and another is the statistical approach. Symbolic approach is used in [1]–[5] and in [6]–[8] statistical approach is followed. Symbolic approach focuses on the pattern or template matching and verb categorization whereas statistical approach focuses on semantic parsing and reasoning.

Semantic parsing is the process of transforming a linguistic input into a well structured and machine-readable representation of its meaning known as the semantic representation as output. The semantic representation can be a string, a tree or an XML document. Fig. 2 shows an example of semantic representation.

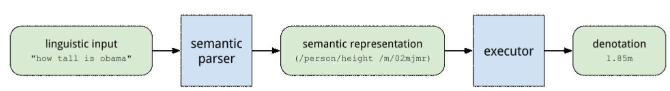

Fig. 2. Example of Semantic Representation [9]

State transition based verb categorization method is used in [6]. Template match technique showed a promising result to solve problems that matched with that particular template in [7]. Both verb categorization and template match techniques are combined in [8] named as **ALGES**.

By semantic parsing, arithmetic word problems are broken up into its constituent parts (Quantity, Containers etc.). From the semantic representation (tree based), a local quantity relationship model and a global equation model are trained. Local

978-1-5386-9111-3/19/$31.00 ©2019 IEEE

quantity relationship model is the probability distribution between numerical quantities of the semantic representation and operators (+, -, x, /). Global equation model is the probability distribution over solutions' correctness.

In this paper, we proposed a weighted probability based the scoring equation of to solve algebraic word problems as in [8] and named after our system as **WALGES**. The predicted probability of an operator between two numerical quantity is the local score and the predicted probability of equations correctness is the global probability. In training, Backpropagation neural network algorithm adjusts the weight to the local score and the global score. In testing, weighted local score and weighted global score give the final probability for a generated equation tree.

To summarize, in comparison to **ALGES** the contribution of this paper are:

- Introduction to a weighted probability based model where weights are adjusted by the neural network algorithm, and
- Our proposed technique named **WALGES** increases the accuracy to predict the solution to the algebraic word problems.

II. RELATED WORKS

We divided related works into three parts: (1) Parsing the arithmetic word problem into its constituent parts (i.e. Nouns, Verbs, Quantities, Containers etc.), and (2) Generating equations using the quantities, containers, and others.

Analyzing the natural language sentence structures and get semantic representation was used in paper [10]–[13]. A verb categorization method for solving elementary level addition and subtraction problems proposed in [7]. To Divide constituent parts like - *entities* and *containers*, semantic parsing was used.

A general structure for the similar type of problem text proposed in [6]. It could map a system of linear equations to two or more unknowns. Various local features and global features from the problem text were extracted to map the algebraic word problem into corresponding linear equations. For being equation template huge, it needed enormous time to find the candidate equations.

Both Verb Categorization and template matching combined in ALGES [8]. Entities and containers extracted like [7]. Local features and global features extracted from the problem text similar to [6]. It solved about 70% of the word problems of addition, subtraction, multiplication, and division.

III. METHODOLOGY

Overall methodology of our system WALGES divided into four different categories:

- **Grounding to Base Qsets:** Semantic parsing to break up a problem text into its constituent parts,
- **Equation Tree (Parse Tree) generation:** Equation tree generated from the parts of semantic representation,
- **Learning:** Trained local and global models based on the extracted features from the generated equations, and
- **Inference:** Performed testing operation from trained systems.

In the following sections, we discussed the detailed methodology with an example in Fig. 3:

> Mike had 34 peaches left at his roadside fruit stand. He went to the orchard and picked more peaches to stock up the stand. There are now 86 peaches at the stand, how many did he pick?

Fig. 3. Example of Algebraic World Problem (Subtraction)

A. *Grounding to Base Qsets:*

A *Qset* or *quantified set* is a set which is parsed from problem text as quantities and their properties (i.e. Noun, Verb, Adj, etc.). Base Qsets are formed from the *entities*, *quantities*, *adjectives*, *locations*, *containers*, and *syntactic* information of the problem text that grounded as in ALGES [8].

After grounding the base Qsets, it was combined with *operator* based on their relations. Reordering the Qsets was the last process of grounding. Fig. 4 shows an example of grounded *Qsets* .

> quantities : 34 86 x
> types : "peach" "peach" "stand"

Fig. 4. Example of Grounding based on Fig. 3

B. *Equation tree (Parse Tree) generation:*

Equation trees were generated with the help of Integer Linear Programming from the grounded problem text. These equation trees were used in both learning and inferencing steps. Most desirable M type consistent trees were selected with the help of global consideration of constraints [8]. Fig. 5 shows an example of equation trees based on Fig. 4.

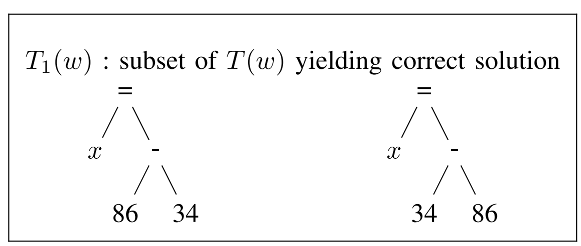

Fig. 5. Example of generated equation trees based on Fig. 4

C. *Learning Process:*

Selection of the best candidate equation tree from the set of equation trees considered the most challenging task. The learning process can be divided into three different sections: (1) *Local Qset Relationship Model*, (2) *Global Equation Model*, and (3) *Weight adjustment to Local and Global score*. Local model was trained for the equation based on the *operators* used between two *Qsets*. Global model was trained based on the final output which could be either *positive* or *negative*.

*1) **Local Qset Relationship Model**:* Local Model distributed the probability over several mathematical operators. For local *Qsets* relationship model, features were selected like [8] where *single qset features*, *relationship between qsets* and *target qset features* extracted using Lin Similarity [14] measurement. Local model was trained using those extracted features. *Target qset* was finalized based on the words such as - *what*, *how*, and *which*. *Operator* is used for labeling the qset features, where *op* $\in \{+,-,*,/\}$. Fig. 6 shows an example of training local model below:

Qsets	Operator
$(86_d, 34_d)$	$-$
$(34_d, 86_d)$	$-$

Fig. 6. Example of training local model Fig. 5

*2) **Global Equation Model**:* Global equation model was the probability distribution over the numerical solution's correctness. For global equation model, features were selected like [8] that were *global lexical features* and *root node features*. Scoring based on the global structure of problem text and tree was trained as a global model. It was generated based on global constraints. Fig. 7 shows the example of global model below:

Training Example	Label
$x = (86 - 34)$	+ve
$(34 - 86) = x$	-ve

Fig. 7. Example of Global Equation Model

In global model, equations were label with either *positive* or *negative* from the numeric solution of the equation. Fig. 8 shows the algorithm for learning process.

*3) **Weight Adjustment to local and global score**:* In the ablation study, ALGES showed that the contribution of global model is superior to score of the local model. Equation. (1) shows our weight based scoring approach:

$$p(t|w) \propto ((\alpha \times \prod_{t_j \in t} \mathcal{L}_{local}(t_j|w)) + (\beta \times \mathcal{G}_{global}(t|w))) \quad (1)$$

Where, α is the weight to the score of *local qset relationship model* and $\beta = (1-\alpha)$ is the weight to the score of the *global equation* model. Multilayer perceptron that implements back-propagation was used to adjust the weight α and β.

D. *Inference:*

For a problem text in inference, a grounded base *qsets* were formed. Then equation trees were generated by Integer Linear Programming. Finally, the based candidate equation tree was selected based on weight to the local and the global model score. Fig. 9 shows the algorithm for inferencing our system used:

Algorithm 1 Algorithm for **Learning Process**

Input: Word Problems, W and Corresponding Solution, L
Output: Local Model, $\mathcal{L}_{local}$, Global Model, $\mathcal{G}_{global}$, and Neural Network, N

for $i = 1$ to n Where, $w_i \in W$ and $l_i \in L$ **do**
  $S \leftarrow$ Base Qsets after grounding and reordering
  $T_i \leftarrow$ Top M relevant equation trees from $ILP(w_i)$
  $T_{l_i} \leftarrow T_i$ where T_i has correct solution, l_i
  Qsets (s_1, s_2) labeled with *op* added to $T_{r_{local}}$ features
  (w, t) labeled with *positive* or *negative* added to global features $T_{r_{global}}$
end for
$\mathcal{L}_{local} \leftarrow$ train a local Qset relationship model by $T_{r_{local}}$
$\mathcal{G}_{global} \leftarrow$ train a global equation model by $T_{r_{global}}$
$N \leftarrow$ training with $\mathcal{L}_{local}$ and $\mathcal{G}_{global}$ as inputs and l_i as the solution
return $(\mathcal{L}_{local}, \mathcal{G}_{global}, N)$

Fig. 8. Algorithm of learning process

Algorithm 2 Algorithm for **Inference**

Input: Word Problem w , Local Qset Relationship Model L_{local} , Global Qset Relationship Model G_{global}, and Trained Neural Network, N
Output: Numeric Solution, l

$S \leftarrow$ Base Qsets after grounding and reordering
$T_i \leftarrow$ Top M relevant equation trees from $ILP(w_i)$
$R_i \leftarrow$ Predicted Probability from N by input $\prod_{t_j \in t} \mathcal{L}_{local}(t_j|w)$,and $\mathcal{G}_{global}(t|w)$
$t^* \leftarrow \arg\max_{t_i \in T} R$
$l \leftarrow$ Numeric Solution from t^*
return l

Fig. 9. Algorithms for inferencing in WALGES

IV. EXPERIMENTS

Experimental setup, dataset description, result and comparison described in the following section.

A. *Experimental Setup:*

For semantic parsing, Standford Dependency Parser in CoreNLP 3.4 [15] used in this experiment. It is the set of natural language technology tools. It is used for parts of speech tagging, name entity recognition, parsing etc. A JSON-RPC server was run publicly at a port. Server returned *'parsetree'*, *'text'*, *'tuples'* containing information about parts of speech, NER, etc. For equation generation from the *qsets* using Integer Linear Programming, CPLEX 12.6.1 [16] was used. For local relationship model and global equation model, LIBSVM [17] was used to train Support Vector Machine (SVM) classifier with Radical Basis Function kernel (RBF Kernel). For local relationship model, gamma parameter of SVM was 0.011 and C was 100. For global equation model, gamma was 0.01 and C was 1000. To solve equations for unknowns, Python's SymPy package was used. For multilayer perceptron, Python's *sklearn* [19] package was used with 5 hidden layers and each of 2 neurons, 0.001 in learning rate, 10^{-8} in epsilon and 200 maximum iteration. For training and testing, 5 fold cross validation was applied. 404 problems were used in training and the rest problems were in testing of total 508 problems.

B. *Dataset:*

Grade school algebraic word problems of single equation was used in this work. Dataset was used in ALGES [8] named as **SINGLEEQ** [18]. Problems and their solutions were collected from *http://math-aid.com*, *http://k5learning.com*, and *http://ixl.com*. 508 problems in total, 1,117 sentences, and 15,292 words was in the dataset. Addition, Subtraction, Multiplication and Division word problems were equally present in SINGLEEQ.

C. *Comparisons and Result:*

We evaluate all the systems on the number of correct answers that report the averages of answer in 5-fold cross validation.

In ALGES [8], final score (predicted probability) for a equation that generate correct solution was calculated by the following 2:

$$p(t|w) \propto (\prod_{t_j \in t} \mathcal{L}_{local}(t_j|w) \times \mathcal{G}_{global}(t|w)) \tag{2}$$

In absence of *Local model* of 2 was referred as *No Local Model* 3 and in absence of *global model* was referred as *No Global Model* 4:

$$p(t|w) \propto \mathcal{G}_{global}(t|w) \tag{3}$$

$$p(t|w) \propto \mathcal{G}_{global}(t|w) \tag{4}$$

In ablation study they showed the accuracy of their system ALGES as in Fig. 10:

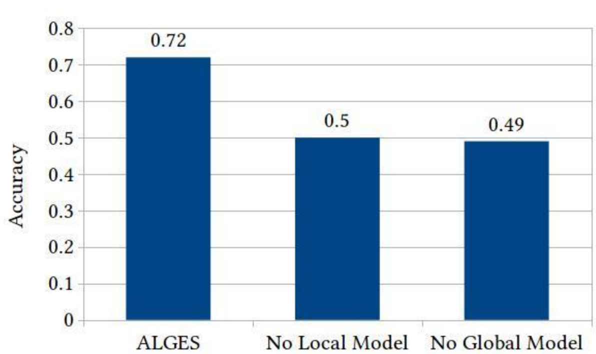

Fig. 10. Ablation Study of ALGES [8]

From this result where *no local model* has higher accuracy than *no global model*, in our system WALGES have considered that weight to local model and global model should vary.

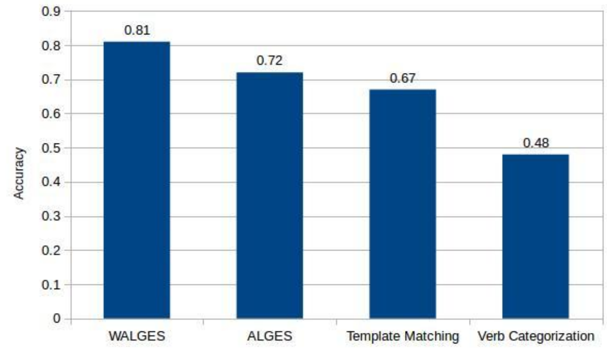

Fig. 11. Performance Comparison

Fig 11 shows that adding weight to local model score and global model score by multilayer perceptron that implies back propagation our system WALGES increases accuracy than ALGES [8], template matching [6], and verb categorization [7] for SINGLEEQ dataset. Accuracy shown by our system WALGES is **81%** which is **9%** greater than ALGES. Our system reduced the error by **32%**.

Dividing the dataset into several subsets of 254, 127 and 63 arithmetic word problems, performance of all the systems are given in I:

TABLE I
PERFORMANCE COMPARISON ON SUBSETS OF **SINGLEEQ**

System	508	254	127	63
WALGES	**0.81**	**0.66**	0.52	0.30
ALGES	0.72	**0.66**	**0.66**	**0.63**
Template-based	0.67	0.60	0.46	0.26
Verb Categorization	0.48	0.42	0.37	0.31

From the performance measure of the I, it shows that our system's performance degrades when the dataset size decreases. But ALGES almost constant accuracy for all the subsets of dataset.

V. CONCLUSION AND FUTURE WORK

In this paper, we used neural network algorithm to generate the solution of algebraic word problems more accurately. By weighted probability based scoring, our proposed system WALGES fixes more error when determining the best candidate equations. But degrading the dataset size, the performance of our system decreases.

Our future work will focus on solving multivariable problems. We would also like to extend our system which could ground the voice input. The code and data for WALGES are publicly available.

REFERENCES

[1] D.G. Bobrow, Natural language input for a computer problem solving system, Report MAC-TR-1, Project MAC, MIT, Cambridge, June 1964.

[2] D.G. Bobrow, Natural language input for a computer problem solving system, Ph.D. Thesis, Department of Mathematics, MIT, Cambridge, 1964.
[3] E. Charniak, CARPS: a program which solves calculus word problems, Report MAC-TR-51, Project MAC, MIT, Cambridge, July 1968.
[4] Y. Bakman, “Robust understanding of word problems with extraneous information,” in http://arxiv.org/abs/math/0701393, 2007.
[5] C. Liguda and T. Pfeiffer, “Modeling Math Word Problems with Augmented Semantic Network,” in NLDB, pp. 247–252, 2012.
[6] N. Kushman, Y. Artzi, L. Zettlemoyer, and R. Barzilay, “Learning to automatically solve algebra word problems,” in Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL), 2014.
[7] M. J. Hosseini, H. Hajishirzi, O. Etzioni, and N. Kushman, “Learning to solve arithmetic word problems with verb categorization,” in EMNLP, 2014.
[8] Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang, “Parsing algebraic word problems into equations,” in Transactions of the Association for Computational Linguistics, vol. 3, pp. 585–597, 2015.
[9] Bill McCarney, “SippyCup,” 2015. [Online]. Available: http://nbviewer.jupyter.org/github/wcmac/sippycup/blob/master/img/sippycup-figure-1.svg 2015. [Accessed: 07-Oct-2018]
[10] D. Gildea, and D. Jurafsky, “Automatic labeling of semantic roles,” in Computational Linguistics, 28(3), 2002.
[11] X. Carreras. and L. Marquez, “Introduction to the CoNLL-2004 shared task: Semantic role labeling,” in Proceedings of CoNLL, 2004.
[12] L. Marquez, X. Carreras, K.C. Litkowski, and S. Stevenson, “Semantic role labeling: an introduction to the special issue,” in Computational Linguistics, 34(2), 2008.
[13] C. Baker, M. Ellsworth, and K. Erk. 2007, “Frame semantic structure extraction,” in Proceedings of SemEval, 2007.
[14] Dekang Lin, “An information-theoretic definition of similarity,” in ICML, volume 98, pages 296–304, 1998.
[15] Marie-Catherine De Marneffe, Bill MacCartney, Christopher D. Manning, et al., “Generating typed de- pendency parses from phrase structure parses,”, in Proceedings of LREC, volume 6, pages 449–454, 2006.
[16] IBM ILOG., IBM ILOG CPLEX Optimization Studio 12.6, 2014.
[17] Chih-Chung Chang and Chih-Jen Lin, “LIBSVM: A library for support vector machines,” in ACM Transactions on Intelligent Systems and Technology, 2:27:1– 27:27, 2011.
[18] Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang, “SINGLEEQ,” 2016. [Online]. Available: https://gitlab.cs.washington.edu/ALGES/TACL2015/blob/master/questions.json. [Accessed: 26- Nov- 2017].
[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et. al., “Scikit-learn: Machine Learning in Python,” in Journal of Machine Learning Research, volume 12, pages 2825–2830, 2011.