JoEICT (Journal of Education and ICT)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/joeict
ISSN : 2987-3215
Vol. 8, No. 2, December 2024, Pp. 45-49

IDENTIFYING ARITHMETIC OPERATION IN MATH WORD PROBLEM BASED ON RECURSIVE NEURAL NETWORK AND SUPPORT VECTOR MACHINE

Agung Prasetya*1)
1. Informatika, Fakultas Sains dan Teknologi, Universitas Bhinneka PGRI, Indonesia

Article Info
Keywords: Math Word Problem, Arithmetic Operation, Recursive Neural Network, Support Vector Machine

Article history:
Received 17 Oktober 2024
Revised 15 November 2024
Accepted 25 November 2024
Available online 1 December 2024

DOI :
https://doi.org/10.29100/joeict.v8i2.7421

* Corresponding author.
Corresponding Author
E-mail address:
agung@ubhi.ac.id

ABSTRACT
Math word problems act as a test bed to design an intelligent system. An approach is needed to identify arithmetic operations including addition, substraction, multiplication and division. Template-based approaches have addressed this problem. However, the template-based approach is less efficient because it requires the process of building a template repository that have to cover a wide variety of story implied by math words. The template-based approach is potentially sub-optimal when solving story problems that have not been covered yet by templates. The proposed approach resolves this by using Recursive Neural Network and Support Vector Machine. Recursive neural network is used as an encoder that can generate semantic vectors of math word problems. Then, this vector becomes as an input for a Support Vector Machine-based classifier. Tests were conducted on a dataset collected manually from Kemdikbud' electronic school books. The results showed that the proposed approach does not require the formation of templates, thereby reducing human involvement. In addition, the use of Recursive Neural Network reduces feature engineering making it more efficient. Experimental results by applying k-fold cross validation show that the proposed approach has an accuracy of 81% and a precision of 66%.

.

I. INTRODUCTION

Math word problems serve as a test bed to determine whether a system is capable of communicating with humans [1], [2], [3]. Math word problems, although simple and limited in nature, model the question-and-answer interaction of machines with humans. To solve math word problems written in natural language, an approach is needed to determine what kind of arithmetic operations are involved.

A number of studies have been proposed to identify arithmetic operation using templates [1], [4], [5], [6], [7], [8]. Template-based approaches need to specify a repository contain a number of templates in advance. Although able to detect the arithmetic operations in the problem, template-based approaches require a large variety of templates covering possibility of stories implied in math word problem. Those templates must be able to cover a wide range of possible forms of story problems. When faced with new problems that have not been covered yet by templates, the template-based approaches have a drawback that be unable to accurately detect arithmetic operation.

Based on the above limitation, this research propose an approach for arithmetic operation detection in math word. Problems are modelled as a classification problem with classes in the form of arithmetic symbols +, -, * and /. Each symbol represents addition, subtraction, multiplication and division operations. To reduce human involvement in feature engineering, this research proposes an encoder-decoder model. More concretely, this research uses a Recursive Neural Network (RvNN) [9], [10], [11] that generates embedding according to the semantic structure of the story implied in math word problem. Furthermore, the embedding vector becomes the input for a Support Vector Machine (SVM)-based classifier [12], [13], [14].

This research proposes RvNN and SVM to detect an arithmetic operation of a math word problem. Both models were trained using the existing dataset in [15], [16], [17]. The proposed approach solves non-mixed problems, i.e. problems involving one type of arithmetic operation. The performance of the proposed approach is measured based on precision and accuracy.

II. RESEARCH METHOD

The proposed approach in this research consists of two models namely RvNN and SVM. RvNN is used as an encoder which embed text to vector. This vector represent semantic of math word problem. The existence of RvNN allows numerical processing of a symbol. While SVM is used as a classifier that takes as an input semantic vector and produces one of 4 types of arithmetic operations.

A. RvNN based Math Word Encoder

Given a text, RvNN attempts to generate a vector. This vector represents the meaning of the text. Two texts are close in meaning if the two vectors have the least distance in an n-dimensional vector space. RvNN is used in this research because it involves less human involvement in extracting features, making it more efficient.

RvNN works by applying the principle of semantic composition [18]. This principle states that the meaning of a unit of text is composed from the meanings of the constituent parts of the text by following a certain grammatical structure. Based on this principle, this research considers the sentence as a unit of semantic of the math word problem and then the word as a constituent of the meaning of the sentence.

The semantic vector of the sentence is obtained through the encoding process. The encoding process begins by modelling the sentence as a binary tree with words occupying terminal nodes and phrases or sentences at non-terminal nodes. Each word in the binary tree is mapped to an n-dimensional vector space using the Word2vec model [19], [20], [21] resulting in a word semantic vector. The vectors at the non-terminal nodes, $p \in \mathbb{R}^n$, are obtained by applying a non-linear activation function, $tanh$ as in [16], [17] and utilising vector at the left child, v_L, and other vector at the right child, v_R

$$p = tanh(W_L \cdot v_L + W_R \cdot v_R + b_e)$$

Where both W_L and W_R are the encoding weights in the left and right child and b_e is the encoding bias and all three are obtained through training. Encoding is done in bottom-up manner recursively to get all non-terminal node vectors until it reaches the root node. The vector at the root node represents the sentence semantic vector.

The vectors in the non-terminal nodes are adjusted by reconstructing p into two vectors, v_L' and v_R' through a linear function as in [9], [17]. This process is called decoding:

$$v_L' = U_L \cdot p + b_d^{(1)}$$
$$v_R' = U_R \cdot p + b_d^{(2)}$$

where U_L and U_R are the decoding weights in the left and right child and both $b_d^{(1)}$ and $b_d^{(2)}$ are the decoding biases and all four are obtained through training. The vector p is adjusted such that the distance between the decoding vector and the vector before encoding, E_{nt}, is minimum:

$$E_{nt} = \|v_L' - v_L\|_2 \; + \|v_R' + v_L\|_2$$

The encoding and decoding process to obtain the sentence semantic vector follows a binary tree. In this research, the shape of the binary tree is based on parse trees [22], [23].

The semantic vector of a math word problem, in principle, are obtained by encoding-decoding as above. The type of structure used as a binary tree is the Rhetorical Structure Tree (RST). Two sentences are involved as left and right child if they have a nucleus-satellite relation. This research uses Max-Margin Parsing [24], [25] for paragraph parsing as in [16].

B. SVM based Arithmetic Operation Classifier

The results of encoding process in the form of vectors represent the meaning of the story problem. This vector can be used as input for the classifier in detecting the type of arithmetic operation. This research only identifies one type of operation in a problem. Problems that contain the same type of operation mean that the vectors are close so that they can be said to have similar meanings.

Given L training data with each data consisting of a D-dimensional input vector, $x_i \in \mathbb{R}^D$ and class $y_i \in \{-1,1\}$ and $i = \{1, \ldots, L\}$, Support Vector Machine (SVM) determines the hyperplane that separates the data into two classes. It is assumed that the training data is not completely linearly separable, so a positive slack variable is used in SVM. For each binary SVM, several values are calculated, namely:

- **H** matrix with each element $H_{ij} = y_i y_j x_i \cdot x_j$ for $i, j = \{1, \ldots, L\}$. Since it is assumed that the data is not linearly separable, the calculation of the dot product, i.e. $x_i \cdot x_j$, in the formation of matrix H is replaced with a non-linear kernel function. This research will trial Radial Basis Kernel (RBF), Polynomial Kernel and Sigmoidal Kernel.
- Selecting the value of parameter C that determines the significance of misclassification
- Determining the Lagrange multiplier α such that $\sum_{i=1}^{L} \alpha_i - \frac{1}{2}\alpha^\top \mathbf{H}\alpha$ is maximum s.t the constraints:

$0 \leq \alpha_i \leq C \; \forall_i$ dan $\sum_{i=1}^{L} \alpha_i y_i = 0$

- Calculating the weights $w = \sum_{i=1}^{L} \alpha_i y_i x_i$
- Determining the set of Support Vector S by finding the indices such that $0 < \alpha_i < C$
- Determining the bias $b = \frac{1}{N_s} \sum_{s \in S} (y_s - \sum_{m \in S} \alpha_m y_m x_m \cdot x_s)$

Each new data x' can have its class determined by calculating $y' = sign(w \cdot x' + b)$

The input vector is the encoding result of RvNN while the class is the type of calculation operation, namely addition, multiplication, division and subtraction. Because the number of classes is more than two, this study uses a one-to-one strategy by running binary SVM for each of 4 classes.

III. Result And Discussion

A. Dataset

The dataset for testing in this study consists of a collection of basic math word problems as in study [17]. Each math word problem contains only one of the 4 types of arithmetic operations, namely subtraction, addition, multiplication and division. All story problems are taken from the electronic school book http://bse.kemdikbud.go.id/ with a total of 210 problems labelled as addition, 230 labelled as subtraction, 220 labelled as multiplication and 210 labelled as division.

B. Result

Following study in [17], the dimension of word vector was set at 200. Tests were conducted to determine the optimal kernel function among RBF, Sigmoidal and Polynomial Kernel. To reduce bias in testing, the test was conducted by applying k-fold cross validation with $k = 6$. All training data were randomly selected for each fold. For each fold, five folds as training and the other folds as test data. The performance of the proposed model is measured based on 2 types of metrics namely accuracy and precision.

The accuracy measurement results for all folds are shown in Table I below:

TABLE I
ACCURACY FOR ALL FOLD

Kernel Function	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Avg
RBF	0.78	0.77	0.69	0.69	0.71	0.73
Sigmoidal	0.82	0.82	0.79	0.79	0.81	0.81
Polynomial	0.72	0.68	0.69	0.68	0.67	0.69

Based on table 1 above, the best accuracy is shown for Sigmoidal kernel function type, which is 0.81. However, the accuracy value is not too far from RBF.
The precision measurement results for all folds are shown in Table II below:

TABLE II
PRECISION FOR ALL FOLD

Kernel Function	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Avg
RBF	0.58	0.61	0.59	0.61	0.59	0.58
Sigmoidal	0.66	0.69	0.68	0.66	0.67	0.66
Polynomial	0.57	0.59	0.58	0.58	0.59	0.57

Based on table 2 above, the best precision is shown for the Sigmoidal kernel function type which is 0.66.

C. Discussion

Based on the test results for precision and accuracy measurements, the type of kernel function affects the performance of the classifier in detecting counting operations. The optimal kernel function in this research is Sigmoidal. This research also shows that the use of semantic vectors obtained without feature engineering can contribute to an accuracy rate of 0.81. Seen from the perspective of human work, the RvNN model contributes in terms of reducing human involvement making it more efficient.

IV. CONCLUSION

Experiments show that the proposed model based on RvNN and SVM is able to detect the type of arithmetic operation with an accuracy of 0.81 and precision of 0.66 for Sigmoidal kernel. Future research can consider detecting the types of arithmetic operations in math word problems that are more than two types. For this case, future research can consider paragraph compositionality.

REFERENCES

[1] S. R. Raiyan, M. N. Faiyaz, S. M. J. Kabir, M. Kabir, H. Mahmud, and M. K. Hasan, 'Math Word Problem Solving by Generating Linguistic Variants of Problem Statements', Jun. 24, 2023, *arXiv*: arXiv:2306.13899. doi: 10.48550/arXiv.2306.13899.

[2] A. M. Kareem and A. Obied, 'Testbed for intelligent agent: A survey', *J. Al-Qadisiyah Comput. Sci. Math.*, vol. 13, no. 2, p. Page-23, 2021.

[3] A. Mukherjee and U. Garain, 'A review of methods for automatic understanding of natural language mathematical problems', *Artif. Intell. Rev.*, vol. 29, no. 2, pp. 93–122, Apr. 2008, doi: 10.1007/s10462-009-9110-0.

[4] L. Wang *et al.*, 'Template-based math word problem solvers with recursive neural networks', in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, pp. 7144–7151. Accessed: Dec. 24, 2024. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4697

[5] J. Qin, Z. Yang, J. Chen, X. Liang, and L. Lin, 'Template-Based Contrastive Distillation Pretraining for Math Word Problem Solving', *IEEE Trans. Neural Netw. Learn. Syst.*, 2023, Accessed: Dec. 24, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10113691/

[6] L. Wang, Y. Wang, D. Cai, D. Zhang, and X. Liu, 'Translating a Math Word Problem to an Expression Tree', Nov. 15, 2018, *arXiv*: arXiv:1811.05632. doi: 10.48550/arXiv.1811.05632.

[7] Y. Wang, X. Liu, and S. Shi, 'Deep neural solver for math word problems', in *Proceedings of the 2017 conference on empirical methods in natural language processing*, 2017, pp. 845–854. Accessed: Dec. 24, 2024. [Online]. Available: https://aclanthology.org/D17-1088/

[8] W. Zhao, M. Shang, Y. Liu, L. Wang, and J. Liu, 'Ape210K: A Large-Scale and Template-Rich Dataset of Math Word Problems', Oct. 09, 2020, *arXiv*: arXiv:2009.11506. doi: 10.48550/arXiv.2009.11506.

[9] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, 'Parsing natural scenes and natural language with recursive neural networks', in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 129–136. Accessed: Dec. 24, 2024. [Online]. Available: https://nlp.stanford.edu/pubs/SocherLinNgManning_ICML2011.pdf

[10] Q. Guo, Z. Yu, Y. Wu, D. Liang, H. Qin, and J. Yan, 'Dynamic recursive neural network', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5147–5156. Accessed: Dec. 24, 2024. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Guo_Dynamic_Recursive_Neural_Network_CVPR_2019_paper.html

[11] O. Irsoy and C. Cardie, 'Deep recursive neural networks for compositionality in language', *Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, Accessed: Dec. 24, 2024. [Online]. Available: https://proceedings.neurips.cc/paper/5551-deep-recursive-neural-networks-for-compositionality-in-language.

[12] D. A. Pisner and D. M. Schnyer, 'Support vector machine', in *Machine learning*, Elsevier, 2020, pp. 101–121. Accessed: Dec. 24, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128157398000067

[13] S. Suthaharan, 'Support Vector Machine', in *Machine Learning Models and Algorithms for Big Data Classification*, vol. 36, in Integrated Series in Information Systems, vol. 36. , Boston, MA: Springer US, 2016, pp. 207–235. doi: 10.1007/978-1-4899-7641-3_9.

[14] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, 'Support vector machines', *IEEE Intell. Syst. Their Appl.*, vol. 13, no. 4, pp. 18–28, 1998.

[15] A. Prasetya and T. A. Cahyono, 'PARSING STRUKTUR PARAGRAF BERBASIS NEURAL NETWORK', *JIPI J. Ilm. Penelit. Dan Pembelajaran Inform.*, vol. 7, no. 2, pp. 608–614, 2022.

[16] A. Prasetya, C. Fatichah, and U. L. Yuhana, 'Parsing struktur semantik soal cerita matematika berbahasa indonesia menggunakan recursive neural network', *Register*, vol. 5, no. 2, pp. 106–115, 2019.

[17] A. Prasetya, 'Deteksi Operasi Hitung Pada Soal Cerita Campuran Menggunakan Recursive Neural Network', PhD Thesis, Institut Teknologi Sepuluh Nopember, 2019. Accessed: Dec. 24, 2024. [Online]. Available: https://repository.its.ac.id/61173/

[18] F. J. Pelletier, 'The Principle of Semantic Compositionality', *Topoi*, vol. 13, no. 1, pp. 11–24, Mar. 1994, doi: 10.1007/BF00763644.

[19] Y. Goldberg and O. Levy, 'word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method', Feb. 15, 2014, *arXiv*: arXiv:1402.3722. doi: 10.48550/arXiv.1402.3722.

[20] Q. Le and T. Mikolov, 'Distributed representations of sentences and documents', in *International conference on machine learning*, PMLR, 2014, pp. 1188–1196. Accessed: Dec. 24, 2024. [Online]. Available: http://proceedings.mlr.press/v32/le14.html?ref=https://githubhelp.com

[21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, 'Distributed representations of words and phrases and their compositionality', *Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, Accessed: Dec. 24, 2024. [Online]. Available: https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html

[22] B. Galitsky, 'Learning parse structure of paragraphs and its applications in search', *Eng. Appl. Artif. Intell.*, vol. 32, pp. 160–184, 2014.

[23] E. M. Sibarani, M. Nadial, E. Panggabean, and S. Meryana, 'A study of parsing process on natural language processing in bahasa Indonesia', in *2013 IEEE 16th International Conference on Computational Science and Engineering*, IEEE, 2013, pp. 309–316. Accessed: Dec. 24, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6755234/

[24] B. Taskar, D. Klein, M. Collins, D. Koller, and C. D. Manning, 'Max-margin parsing', in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004, pp. 1–8. Accessed: Dec. 24, 2024. [Online]. Available: https://aclanthology.org/W04-3201.pdf

[25] A. Bauer, S. Nakajima, N. Görnitz, and K.-R. Müller, 'Optimizing for measure of performance in max-margin parsing', *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2680–2684, 2019.