Received: November 16, 2021. Revised: December 28, 2021. 200

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

# Automatic Assessment of Answers to Mathematics Stories Question Based on Tree Matching and Random Forest

**Umi Laili Yuhana[1]* Vessa Rizky Oktavia[1] Chastine Fatichah[1] Ayu Purwarianti[2]**

[1]*Department of Informatics Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia*
[2]*School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung, Indonesia*
* Corresponding author's Email: yuhana@if.its.ac.id

**Abstract:** Math word problems can be solved with a good understanding of language, correct translation, and use of proper operations. However, elementary school students will only get the correct answer if the result is done correctly. Then a numeracy competency detection system is needed through the answer of math story questions. This study aims to build a system of checking student answers in stages. The main contribution is the technique for comparing the trees from two multimodal input and assess student answer automatically. System extracted operand from math story sentences and classified operator using random forest to generate the key then convert it to the tree. We use OCR library to extract text from student's answer image and identify operand, operator, and result to build student's answer tree. A tree matching is applied to compare the similarity of trees for automatic assessment. The dataset used in this research is 500 questions, 300 data for training, and 200 data for testing. There are two categories of questions, single and mixed operator, with five class namely addition, subtraction, multiplication, division, and mixed. Based on the experiment, the accuracy of classification for mixed operator is 68.8%, whether for single operator is 84.31%. For tree matching, we achieved 78.12% accuracy with the error occurs due to the limitations of processing student's answer image.

**Keywords:** Automatic assessment, Word problems, Student competencies, OCR, Random forest.

## 1. Introduction

Artificial intelligence or what is often referred to as Artificial Intelligence (AI) is a technology that is not endlessly developed by humans. The creation of AI itself cannot be separated from the human desire to ease the work while at the same time helping the continuity of activities in daily life. In this 5.0 Society era, AI has reached the stage of taking over human work so that automation is closely linked to the main goal of AI development. Automation using AI is certainly widely used in every aspect of human life, one of which is in the field of education.

AI products that are widely used today are applications that can solve mathematical problems automatically, by entering math problems in the form of images. The use of technology from this kind of application should be developed not only for students, but also for teachers. If with this application, students can solve problems quickly, teachers should also have facilities to be able to correct students' answers quickly too. Checking student answers will certainly take a long time if it is done manually. Therefore, we need an application that can evaluate student answers automatically. This technology not only help teacher but it can be used to aid student evaluate his competency independently.

In the field of education, there are many competencies that require a process as a material for consideration of the output. For example, when working on an easy problem, namely addition, students can easily answer it. However, in mixed questions that have higher complexity, student competence cannot be judged solely from the final answer. The teacher must consider the existing process in working on the problem so that it can see where the student's mistakes are. For example, in

addition to being followed by subtraction, it turns out that students are only wrong in the addition section. Thus, the teacher will understand that the competence that needs to be improved from the students is the summation part. Finally, students can be given more direction and practice questions regarding addition so that in the future students can solve similar problems correctly.

The process of solving the answers to questions will be even more complex if the questions are presented in the form of stories. In addition to requiring more understanding to look at every word in the problem, students are also required to translate the story into mathematical sentences. If the story problem contains more than one operation, of course it will be more difficult for students. So far, students are only judged based on the answers they give, which are true or false. This result represents a number between 1 or 0, nothing in between because it only considers the final answer. To investigate further about the actual competence of students in solving math story problems, it would be nice to consider each process of working on the problem. Therefore, we need a system that can generate answers from math story problems that not only produce the result, but also include steps to solve the problem.

Research in the field of math story problems begins with the analysis that math story problems have three problems, namely language, numeracy, and the relationship between the two [1]. This research fully discusses the category of math story problems and the sub-problems in it. In addition, it also discusses the structure of the math story problem itself and what are the factors that cause a person's success in solving story problems or hinder the search for solutions to the problem. In general, this research was conducted using math story problems in English.

In 2015 at the University of Washington, research was conducted on how to convert mathematical story problems into the form of mathematical equations. The result of this research is a tree that represents each operator and operand in the math story problem [2]. In the same year, a study in China proved that not only mathematical story problems that can produce trees, but also math problems that do not contain stories or commonly called Word Problems (WP). The questions used have a higher level of complexity than elementary school math story questions and the analysis in this study is more directed at the semantic structure of the questions [3]. In 2016, research was conducted in Indonesia regarding the difficulties experienced by elementary school students in solving math story problems [4].

Then, in 2018 a study was conducted that resulted in a robot that can solve simple math story problems where the story problem only has one type of operation [5]. Robots that have been the robot that is made solves mathematical story problems symbolically and does not consider the semantic element because the problems that are solved have low complexity. Furthermore, in research conducted in India, it was stated that unlike natural language programming (NLP), artificial intelligence (AI), and machine learning (ML), research in the field of mathematical story problems was not widely carried out [6]. The latest research was conducted in Indonesia, considering the semantic structure of math story problems and inserting it into a recursive neural network [7]. The results of this study are which operators are used in a story problem but have not solved the problem itself automatically. It is essential to build assessment system with ability for generating key answer of math story problem and assess the student answer automatically.

This research proposes the technique for comparing the trees from two multimodal input (text of question and image of student answer). The expected contribution of this research is to compare the tree built from the best classification method with the tree results from student answers. From this tree comparison, it will be seen which part of the student's competence should be further developed. The paper is organized as follows. Section 2 discusses related work in the field of math story research and technique. Section 3 describes a brief overview of the proposed method for tree converting and matching. Section 4 presents result and discussion. Finally, Section 5 provides our conclusions.

## 2. Related research

Math Story Problems are problems that are presented in a narrative which contains mathematical operations in it. An example can be seen in Fig. 1. The problem consists of three sentences, each of which contains different information. The first sentence contains a number which is the beginning of the problem and there is also the word apple which is the unit that is the focus of the story. The second sentence contains another number with the same units but can only be seen implicitly and not directly written as the same unit, namely apples. Then in the last sentence there is a question that refers to the two previous numbers. Each story problem usually has something in

Bimo has 3 apples. Then 2 were given to Wiwik. How many apples do Bimo have left?

Figure. 1 Simple math problem

common like this example but has very different variations.

The difficulty level of the story questions is very diverse and has many factors. The three main factors in math story problems are language factors, mathematical factors, and general factors [1]. Language factors can be divided into two, namely structure and semantics. Structurally, a math story problem has complexity in terms of quantitative, vocabulary, and placement or discussion of questions. In quantitative terms, it includes the number of letters, words, and sentences used, and the proportion of the number of words that contain high complexity. Then on the vocabulary side, the level of difficulty can change along with the use of prepositional phrases, passive words, and clause structures. Finally, the placement of question words also has its own complexity because the place can vary from the front, middle, or back.

Meanwhile, semantically it can be divided into five, namely verbal linguistic cues, the use of phrases in the sign, the concept of restatement, semantic relations and objects, and the presence of distractors. Verbal linguistic cues are the delivery made by story problems which indicate that in the problem there is a problem to be solved. Then the use of phrases in gestures means not only using sentences but also phrases when conveying the problem to be solved. Next is the concept of re-discussion, meaning that when digesting a mathematical story problem, then to solve it we must discuss it in its mathematical form. Then, the semantic relationship with the object is which word, phrase, or sentence indicates the existence of a mathematical operation there that is related to the operand in the story problem. Finally, the existence of a distractor or sentence that has nothing to do with the mathematical form of the story problem but is often included just to divert attention.

In terms of mathematical factors, structurally there are two things that can be considered, namely the number property (number of digits and number type) and the type of operation used. Semantically, it is divided into two, namely mathematical strategies in finding solutions (counting from the largest number and number of combinations of numbers) and relevance of information (representation in mathematical form). Meanwhile, the factors in general are on social and skill aspects, categories of questions, strategies in finding solutions, and other aspects such as the type of perspective and knowledge of vocabulary.

The research begins by making the semantic structure of a math story problem and then representing it in the form of a tree [3]. Each word is mapped based on its type and then a tree will be built according to the order of words in the problem. This research was conducted using English story questions and stopped until a tree was formed and got an F1 measure value of 73.8%. Then, another research was also conducted, namely changing the form of math story questions into their mathematical form[2]. The conversion of mathematical story problems into mathematical operations uses Integer Linear Programming (ILP) which effectively translates problems that only contain addition, subtraction, multiplication, and division operations. However, there are limitations in this study because it has not been able to solve story problems automatically and even though the accuracy is 84%, it is done using English questions.

Furthermore, in 2018, research was conducted which was a development of previous research [3] by adding Logic Inference to the tree formation [8]. The formation of a tree as shown in Fig. 2,

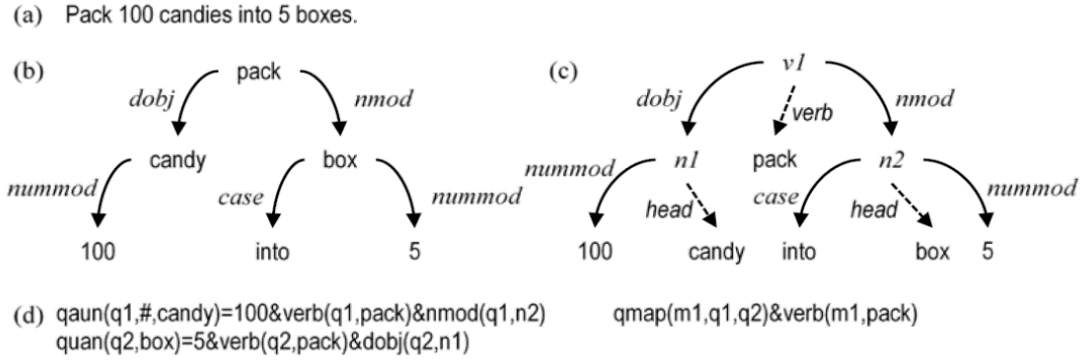


Figure. 2 Examples of logical transformations

includes solving the problem into word types and then determining the head of the tree that is built. This study has the advantage that it covers more types of questions, but the drawback is that there are still some errors that occur when encountering ambiguous questions.

Then, in 2019 in Indonesia, research was conducted that was more focused on the semantic structure of math story questions. The semantic structure of this question is then entered into a binary candidate tree whose weight will then be calculated using a recursive neural network (RvNN) [7]. This research results in what types of operations are contained in a math story problem, or rather detects whether a math story problem has one or more types of operations in it. From this state-of-the-art research on math story problems, a system can be developed that not only can answer automatically, but also shows the stages of problem solving.

### 2.1 Convert math story problems to tree

Mathematical story problems are identical to the formation of trees in their solutions. When constructing a tree, each operand can be a child and each operator can be a parent of a tree as shown in Figure 3. Of all the existing research, the latest research specifically dedicated to discussing trees in story problems was carried out in 2018 and 2019. In 2018, this research resulted in a framework that was specifically built to convert math story questions into tree form [9]. The results of this study only reached the formation of a tree, not yet finished solving story problems based on the tree that was built. Then in 2019, a study was conducted that considered the sequence or sequence of the given math story questions to reduce errors in problem solving. The tree that is built uses a bottom-up approach and has a sequence that is arranged based on the priority of number operations. So, if there is multiplication, it will take precedence according to the number operation conditions [10]. By considering the order of operations, it will produce the right answer and in accordance with the arithmetic calculations of the story problems. This study uses a Bi-directional Long Short Term Memory Network (Bi-LSTM) encoder. The first is to convert the sentences of math story problems into vectors as shown in Eq. (1). Each word denoted by $t$ will be included in the vector $e_t$. Then, this vector will be the input of the Bi-LSTM model which is built with Eq. (2) where $h_t$ is the result of the concatenation of $h_t^f$ and $h_t^b$ which is the hidden layer of forward propagation and backward propagation.

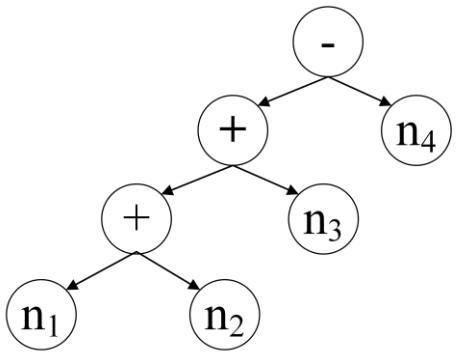


Figure. 3 Example of a tree built from math problems

$$x = \{x_t\}_{t=1}^{n} \quad (1)$$

$$h_t = \left(LSTM\left(h_{t-1}^f, e_t\right); LSTM\left(h_{t-1}^f, e_t\right)\right) \quad (2)$$

### 2.2 Random forest

The Random Forest classification method is a method that is often used for large amounts of data including image data [11-13]. The number of trees or the depth of the tree level used can affect the accuracy of this method. The more trees used; the better accuracy will be obtained. Because the story problem uses trees to solve it, Random Forest is a suitable method for classification because the determination of candidate trees can be made from the votes made during the Random Forest classification.

In mathematical story problems, the dataset for classification is obtained from each part of the problem that contains at least one operation. Each question will be broken down and grouped according to its operation to become a training and testing dataset for the Random Forest classification. The classification used refers to research in 2014 concerning Random Forests which can classify short texts, because in math story problems, usually the text served is not too long [14].

This method is used to build a decision tree consisting of root nodes, internal nodes, and leaf nodes by taking attributes and data randomly according to applicable regulations [15]. The root node is the node located at the top, or commonly referred to as the root of the decision tree. Internal node is a branching node, where this node has at least two outputs and only one input. While the leaf node or terminal node is the last node that has only one input and no output. The decision tree begins by calculating the entropy value with Eq. (3) as a

determinant of the level of attribute impurity and the value of information gain obtained from Eq. (4).

$$E(Y) = -\sum_i p(c|Y)\log_2 p(c|Y) \quad (3)$$

Where $Y$ is the set of cases and $p(c|Y)$ is the proportion of $Y$ values to class $c$, $E$ as entropy.

$$Information\ Gain\ (Y,a) = E(Y) - \sum_{v\in Values(a)} \frac{|Yv|}{|Ya|} E(Y_v) \quad (4)$$

Where Values $(a)$ are all possible values in the case set $a$. $Y_v$ is $a$ subclass of $Y$ with class $v$ corresponding to class $a$. $Y_a$ are all values corresponding to $a$.

### 2.3 Case folding

Every word in the story problem can be a feature. If there are words that are the same but have differences in the use of capital, then these words will be two different features. For that, it is necessary to do case folding first so that all the same words can become one feature. Case folding is changing all letters in a word or sentence to lowercase. If a word contains uppercase letters, either in front, in the middle, or behind, then these letters will be changed to all lowercase letters. This step is often ignored if the classifier or input data does not use "case sensitive" characters. Case sensitive is a condition where the program will assume the same all letters that are input, both uppercase and lowercase letters.

For this research, the program still pays attention to the difference between uppercase and lowercase letters. To uniform the input, it is necessary to have a folding case first. For example, in Fig. 5, there are three words that are the same, but have different use of capital letters. If case folding is not done first, then these three words will become three different features that affect the level of similarity between the two documents.

If a document contains the word “purchased”, then this document will be deemed to bear no resemblance to the document containing the word “*Purchased*”. However, if case folding has been done, then these two words will become the same word, namely "*purchase*". So, documents that have this word will be considered to have a high similarity with other documents that have many of the same words.

Case folding can be done at once in a story problem, done per sentence, or per word. In this study, the authors do case folding per word in each story problem. A story problem is assumed to contain two to four sentences. Each sentence contains numbers and words. All the words in this sentence will alternately go through the loop process and experience case folding or changes to lowercase letters.

Purchasing → purchase
PURCHASED → purchase
purchase → purchase

Figure. 4 Lowered case process

### 2.4 Tokenizing

Words are the basis that will feature in text classification. It is necessary to break a sentence into words to take every word in the story problem. A sentence cannot simply be broken down into a word. Prior to tokenizing, case folding, or punctuation removal is usually done. The stages in these three processes can be carried out sequentially as in this study, but also not sequentially. After we do case folding, all the letters in the story will be lowercase. However, all the words in the problem are still several sentences which must be broken down first.

The separator between one word and another is usually a space or other punctuation mark. In this study, it is considered that the separator between words in a sentence when tokenizing is a space mark. A story problem will go through a loop where in this loop each word will be separated based on the space in the sentence. This process will produce the words that are accommodated in an array. These words will later be transformed into a vector which will be the input of the classification process. In addition to words, the number of words in each sentence will also be counted and become a feature of TF-iDF. In this process, because the numbers have not been separated into other variables, the numbers will be treated as words as well.

## 3. Method

This research carries out a system that can solve math story problems automatically. Starting with pre-processing the story questions then classifying them according to the operator used using the best classification method. We investigated Random Forest and Support Vector Machine (SVM) to find the best performance. After that, build a tree that represents each operand and operator in the problem and then solve the problem. At the same time, the system can also show the competence of elementary

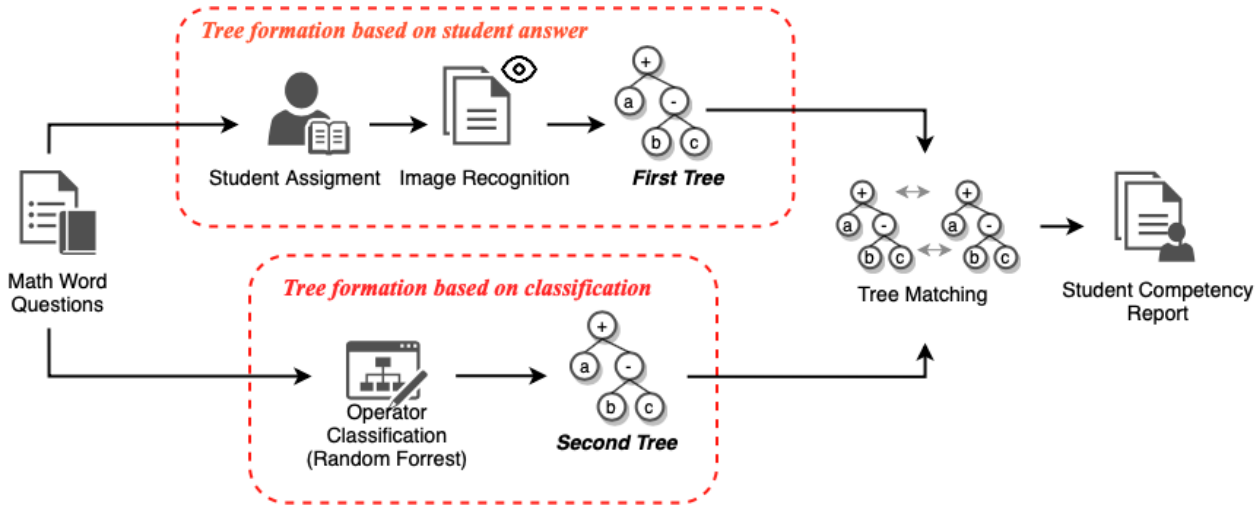


Figure. 5 Method

school students based on their answers when solving math story problems. Questions will be taken in the form of images which will be entered into the system using the OpenCV[1] and Tesseract[2] libraries.

Then, the same as before, a tree will be formed based on the results of the student's answers and will be compared with the tree that has been generated by the previous classification. Comparison of these two trees is done from the left child then proceed to the parent. The result of the comparison of these two trees is that the answers from students will be assessed based on the stages. So, it is not only the final answer that is judged but also every step in the process. The test will be carried out in three scenarios, namely the classification method test, the matching tree result test, and finally the student answer value test based on ground-truth.

Several research questions were prepared, including: How to generate answers automatically from math story problems? How does the Random Forest method perform in classifying operations in story problems based on accuracy? How to form a tree that is used as a comparison between the results of the system and student answers? and how to measure the performance of the proposed system?

Furthermore, the research is limited to the following things: a) story questions with a maximum number of arithmetic operations of two operations. b) story questions with the number of variables that are not known or asked as much as one variable. c) questions that use basic arithmetic operations (+, -, ÷, ×). d) The dataset used is text data and image data with a pixel size of 512 x 512. e) story questions that have a level of difficulty according to Elementary School Grades 1 to 3.

This section shows the process of how the flow of how math word questions are processed into a tree form which will later be used as an indicator of student competency achievement. for the flow of the method can be seen in Fig. 5.

The process begins by providing a dataset in the form of a math word problem that will go through two processes. The first branch is classification. Based on experiment with Random Forest and SVM we found that Random Forest showed the best performance. Therefor we use the Random Forest method to determine the number of operations contained in each part of the story. Then a tree will be created which is taken from how to solve the story problem. This tree is structured based on conditional sentences that contain rules in order of mathematical operation.

While the process in the second branch is to give the math word question to students, then a scanner will be used to take answers from students and convert them into text using the Tesseract library. From the students' answers, a tree will be formed which will then be compared with the tree from the first branch. The result will show how students answer the question based on its mathematical order.

### 3.1 Random forest classification

First, the system will read a mathematical word question which is assumed to contain an operator and two numbers. afterward, the system will do pre-process including tokenization (splitting sentences into words), removing punctuation marks, and changing all letters to lowercase. the process for classification using random forest can be seen in Fig. 6.

From the words contained in the sentence, the system will first detect the numbers in the problem

[1] https://opencv.org
[2] https://github.com/tesseract-ocr/tesseract

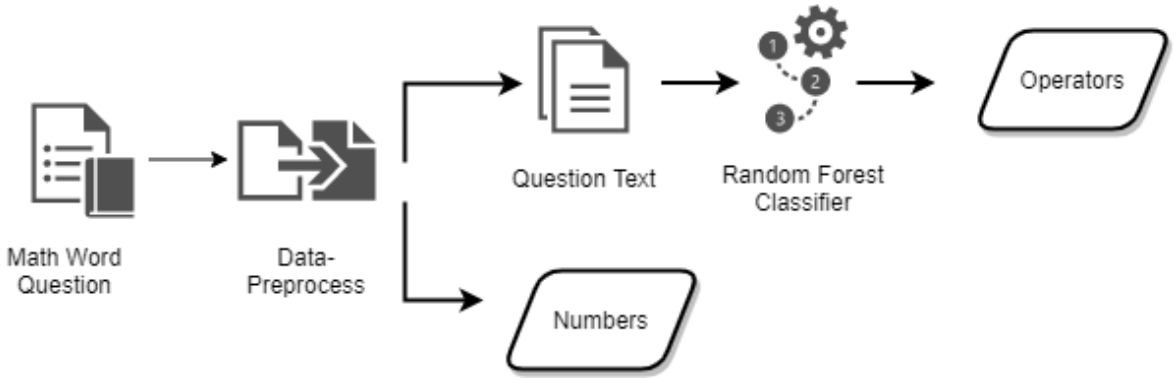


Figure. 6 Question classification using random forest

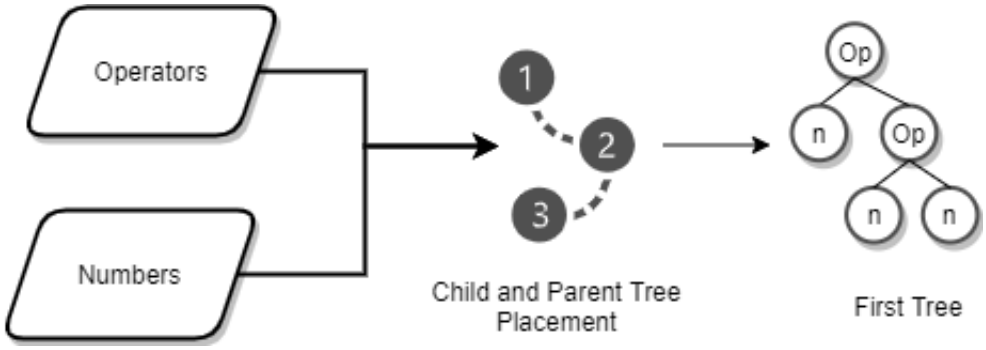


Figure. 7 Tree formation based on classification


Figure. 8 Tree formation based on student answer

and then enter all the numbers into an array. Then, the system will convert the words in the sentence into a TF-iDF vector. This vector will be the feature input on the Random Forest classifier that results in a form of the type of an operator that used in the problem. These numbers and operators will be the input for the next process.

### 3.2 Tree formation based on classification

The input numbers and operators obtained from the previous process will then be built into a tree. The first tree design diagram can be seen in Fig. 7.

The first step is to specify the operator to be a root in the tree. Then, this root will be associated with two children which are numbers or operands. From this process, a tree is formed containing one root parent with two children node.

### 3.3 Tree formation based on student answer

In this process the tree is created based on student answers which are processed from the images of student answers, then from the processed results, there are operators and numbers which are then formed into a new tree. for the process of making the tree can be seen in Fig. 8.

Tree is obtained by translating the picture of students' answers as in the example in Fig. 3.5 into text. After that, a tree will be made according to what the students wrote. Each of the operators will be placed as a parent, each number that will be operated will be a child of each parent. Each

question has at least a root, which is an operator and has 2 children each for each root. The second tree formation can be seen in Fig. 9.

This tree will then be compared with the previous tree to determine whether the student's answer is correct or not. Each node in the tree can have a certain weight which will affect the value obtained by students. Thus, student scores do not only come from the final answer or not, but also through the process of solving math problems. An example of the tree can be seen in Fig. 10.

### 3.4 Tree matching

Tree matching is done in a bottom-up way, so it will start from the leftmost leaf then move up to the parent and so on to the right as shown in Fig. 11. By doing a match like this, it will be more visible the sequence of students' work in solving problems and student errors are also more visible.

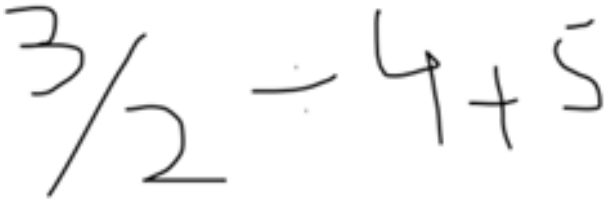

Figure. 9 Example of student answer

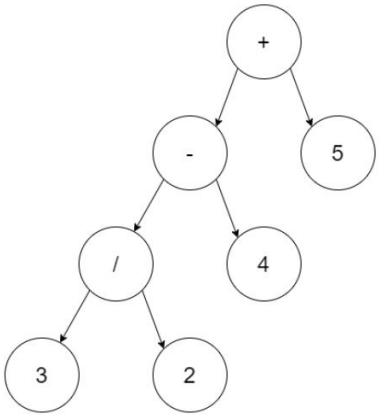


Figure. 10 Example of tree from student answer

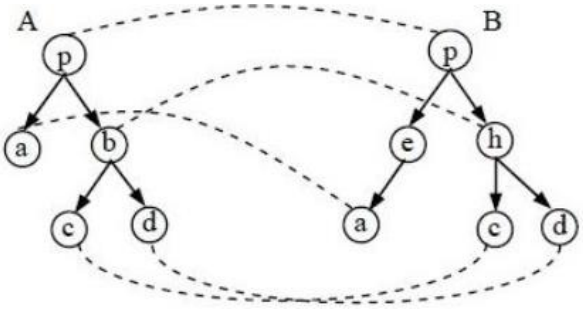


Figure. 11 Botom up tree matching from first tree (A) and second tree (B)

### 3.5 Classification method experiment

In this Classification Method Experiment, it will be done by dividing the test data into two, namely questions that have more than 1 operator and those that have only 1 type of operator. The operators used in the first experiment are addition, subtraction, multiplication, and division. Each operator represents one class each. As for the second experiment, one more class was added which had more than one operator which was a mixture of two operators from different classes.

### 3.6 Tree matching experiment

The tree obtained from the classification will be used as a benchmark to test how the tree formed from the results of students' answers. So, the similarity value will be obtained from each node in the tree. However, because the contents of each node are numbers or operations, the comparison between the two trees will be simpler. Matching is done by using string matching on each leaf as shown in Fig. 12. So, the results obtained are the similarity of the tree at each node. If there are different results, it can be considered that the student's way of solving is not following the ground-truth.

## 4. Results and discussion

The dataset used in this study was taken from datasets conducted in previous studies[7]. The math story questions used come from the Elementary School Package Book with a level of complexity that is adapted to grades 1 to 3 of Elementary School. The data is used as training and testing for classification. Data can come from the public or the author.

The number of datasets consists of 500 data representing each class as much as 100 per class (addition, subtraction, multiplication, division, and mixture). Meanwhile, for the student answer dataset,

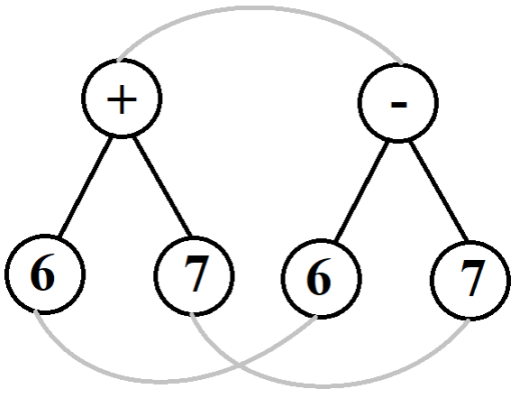


Figure. 12 Example of tree from student answer

```
[0.         0.         0.63144608 0.         0.
 0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.54830459
 0.         0.         0.         0.         0.54830459
 0.         0.        ]
```

Figure. 13 Sample of TF-IDF vector

an image size of 512 x 512 pixels will be used. Student answers were taken from 32 students who each worked on different questions. A total of 13 students were in grade 1 of elementary school, 10 students were in grade 2 of elementary school, and 9 students were in grade 3 of elementary school.

Dataset needs several steps start from processing includes pre-processing, feature extraction, and data sharing into testing and training. The pre-processing of the dataset used in this study includes the omission of punctuation marks only, because in each completion, other pre-processing will affect the classification results (stemming and removal of stop words). Previously, each sentence would be grouped into 4 categories, namely addition, subtraction, multiplication, and division. The assumption given is that each sentence only consists of one type of operation, and each question can have more than one type of operation.

The existing data will be broken down into words and then calculated the Term Frequency – Index Document Frequency (TF-IDF) for each word. Then a vector is formed as shown in Fig 13 Sample of TF-IDF Vector which will be used as input for the classifier. Each column represents each existing term, and each row represents an existing document.

In text pre-processing, 500 raw data have been prepared as input. This data represents 5 classes, each of which represents an operator in mathematics, except for the last class which is a mixture of two operators in other classes. Details of the number of each dataset are in Table 1. All data are made to have the same amount, because data imbalances will result in less-than-optimal results at the time of classification.

For the pre-processing step, it is done by case folding first. Then followed by tokenizing to break down all sentences into word by word. And lastly, followed by punctuation removal to remove unnecessary punctuation marks in sentences.

Table 1. Data-set amount & class

| Class | Amount |
|---|---|
| Addition | 100 |
| Substraction | 100 |
| Multiplication | 100 |
| Division | 100 |
| Mixed Problem | 100 |
| **Total** | 500 |

### 4.1 Classification method with 1 type of operator

The experiment was carried out by dividing the dataset into two parts, namely testing and training. The portion used in the test is 60% data for training and 40% data for testing. The experiment was not carried out using 10-cross validation, but 10 trials with a different number of datasets. This is done so that the testing data used is 40% of the dataset so that it is not too small compared to 10-cross validation which only uses 10% of the dataset.

Using Random Forest, the experiment was carried out 10 times and each trial would have random testing and training data (chosen at random). Each experiment will produce an accuracy value which will eventually be averaged and is the result of the Random Forest classification accuracy. The results of the classification can be seen in Table 2. Table 2.(a) shows accuracy in Random Forest, while Table 2.(b) presents accuracy using SVM. The best accuracy is performed by Random Forest with average of 84.31% and the best accuracy on 86.25%,

Table 2. Accuracy of operator classification using 4 classes: (a) Using random forest and (b) Using SVM

(a)

| Experiment | Accuracy |
|---|---|
| 1 | 83.75% |
| 2 | 84.37% |
| 3 | 82.50% |
| 4 | 85.00% |
| 5 | 83.12% |
| 6 | 83.75% |
| 7 | 85.00% |
| 8 | 86.25% |
| 9 | 84.37% |
| 10 | 85.00% |
| **Mean** | 84.31% |
| **Max** | 86.25% |

(b)

| Kernel | Accuracy |
|---|---|
| Linear | 82.50% |
| Poly | 48.75% |
| Rbf | 71.25% |
| Sigmoid | 75.00% |
| **Best Kernel (Linear)** | 82.50% |

which can be achieved because the dataset used is not stemmed first. Stemming is not done because it will cause several features to be combined, even though these features can be a very important differentiator for each class.

Examples of some features that might reduce accuracy if stemming is done are those that have the words "*take*", "*give*", and "*given*". These words often appear in addition and subtraction problems but have a different use of affixes.

For example, in addition you will usually find the word "*given*". On the other hand, this word reduction is found to have the form of the word "*given*". If stemming is done, then these two words will only become the word "*give*", so that initially two very different features and can be important features for each class, will only become a feature whose function may be almost the same as stop words because they appear too often in the question.

To analyse the first experiment, it can be seen in terms of accuracy that the system produces an average accuracy of 84.31%. SVM achieved the best accuracy using Linear Kernel on 82.50%. Based on this result, we investigated each class for precision and recall Random Forest performance.

### 4.2 Analysis of experiment results with 4 classes

In addition to looking for accuracy results, experiments were also carried out by looking at the value of precision, recall, and F-1 measures of each class when classified. This result is the average of 10 trials that have been carried out. These results can be seen in Table 3.

Precision indicates how much data in the correct class is divided by the total data classified as that class. Suppose an addition class is correctly classified with 90 data, then the system classifies 110 data as an addition class. Then the result is 90 divided by 110 which means the precision value is 0.81 or 81%.

While the recall value is the number of class members that are correctly categorized as that class divided by the number of class members that should be correctly categorized. So, suppose there are 93 data in the subtraction class that are correctly categorized into the subtraction class. If the total reduction class data should have 100 data, then the recall can be calculated as 93 divided by 100, which is 0.93 or 93%.

As for the F-1 measures, it is a weighted comparison of the average precision and recall. The method of calculation is by multiplying the precision value by recall first, then multiplying this value by 2. After that, the result of the sum of precision and recall is also sought. If so, then the number obtained by multiplying by 2 is divided by the number that is the result of the addition.

In the resulting F-1 measures, the 1st and 2nd classes which are addition and subtraction classes reach a value of 90% or more because when looking at the existing features, these two classes have characteristics that tend to be different from other classes. Thus, when classifying, Random Forest can effectively separate these two classes from the other two.

For the 3rd class, the F1-measures value produced is the lowest among the other classes, namely 77%, but still tends to be good because it is still above 60%. In the analysis of the dataset, some of the word features in this 3rd class are somewhat like other classes. In multiplication, it is usually told about someone who has or will give something to someone else, which if there is a word that is like another class it will affect the number of features.

This is a limitation that occurs when classifying story questions, because we cannot specify whether a word is an important feature or not other than its number when using the TF-IDF feature.

Table 3. Performance of operator classification for 4 classes using random forest

| Class | Precision | Recall | F-1 |
|---|---|---|---|
| Addition | 0.92 | 0.89 | 0.91 |
| Substraction | 0.88 | 0.93 | 0.90 |
| Multiplication | 0.77 | 0.77 | 0.77 |
| Division | 0.83 | 0.79 | 0.81 |

### 4.3 Classification methods with mixed class

The experiment was carried out in the same way as before. The results of the classification can be seen in Table 4. For this experiment, the class used was added by one to a total of 5 classes with

Table 4. Performance of operator classification for mixed class using random forest

| Experiment | Accuracy |
|---|---|
| 1 | 68.00% |
| 2 | 69.50% |
| 3 | 70.00% |
| 4 | 70.00% |
| 5 | 69.00% |
| 6 | 68.00% |
| 7 | 68.50% |
| 8 | 69.00% |
| 9 | 67.50% |
| 10 | 68.50% |
| **Mean** | 68.80% |

the last class being a mixed class. This class has a dataset containing two different types of operations. The operations contained in each dataset are a mixture of operations from the addition, subtraction, multiplication, and division classes. The classification results show that the accuracy obtained decreases by 16% when compared to when using 4 classes.

This is clearly influenced by mixed classes which have similarities with almost all classes. This is because in the mixed class, there is more than 1 type of operator even though the system cannot classify this mixed class into two classes at once. The data used contains sentences that are similar to the addition, subtraction, multiplication, and division classes. Because of this similarity, classifiers have difficulty grouping these mixed classes into a separate class. This results in an increase in the error rate when classifying story questions. Therefore, if you want to continue research in this class, there must be another way to do further research, especially for mixed classes only.

### 4.4 Analysis of experiment with 5 classes

In addition to accuracy, a search for precision values, recall, and F-1 measures was also carried out. From here, the main cause of the decrease in accuracy may not seem too bad at first. From Table 5, the recall value for mixed classes has a value of only 29% and F-1 measures 44%. When viewed from the precision side, the fifth class, which is the mixed class, does have a higher precision value than the others. This indicates that when the classification is carried out, of all the data that is considered a mixed class by the classifier, 93% of its contents are indeed members of a mixed class.

However, if you look at the recall value which only managed to reach a value of 29%, then the system still has not classified this mixed class well. This value means that out of 100 mixed class members, only 29 story questions have been successfully grouped into mixed class. The remaining 71 mixed class members were scattered and classified into the wrong class. From this case, it can be concluded that this mixed class greatly affects the accuracy of the other classes, so for this study, only 4 classes are sufficient, namely addition, subtraction, multiplication, and division.

In this mixed class, the F-1 measures that were obtained were only 44%. When compared with the other four classes, the values are very different. In fact, the value can't even reach 50%. For this reason, the classification for this mixed class was not continued as a classifier for tree matching trials. In tree matching, it is assumed that the result of the classifier is true. So, if we use a classifier that produces inaccurate results, the tree matching system will also produce inaccurate results. Due to these considerations, the author finally decided to use a classifier for only 4 classes which resulted in F-1 measures above 70% for each class.

Table 5. Performance of operator classification for 5 classes using random forest

| Class | Precision | Recall | F-1 |
|---|---|---|---|
| Addition | 0.92 | 0.89 | 0.91 |
| Substraction | 0.88 | 0.93 | 0.90 |
| Multiplication | 0.77 | 0.77 | 0.77 |
| Division | 0.83 | 0.79 | 0.81 |
| Mixed Problem | 0.93 | 0.29 | 0.44 |

### 4.5 Tree matching experiment and results

The second experiment was carried out by making a tree from the classification results and a tree from the tesseract scan results. The tree that is built from the results of the classification is made by first classifying a story problem. From the results of the classification, the operators used in the story problems are obtained. Then, all the numbers in the problem will be separated into variables which will later become children of the operator that acts as the parent.

In the tree matching test, there are two trees built. The first tree is a tree resulting from the classification of questions entered by the user into the system. Then, the second tree is the result of students' answers to the question. For this test, all system answers are assumed to be correct and will be considered as ground-truth.

Meanwhile, the matching performance will be analysed depending on how to match the tree and add points to the student scores that will be issued by the system. The maximum score that can be obtained by students is 100 points for each question. The system can assess one question at a time and starts with the user entering a story question.

From the experiment, 25 of 32 tree answers are correctly matching. There are five images of answer unextracted and 2 images unrecognized by tesseract. Based on it, we obtained 78.12% accuracy for tree matching with the resulting error occurs due to the limitations of tesseract in scanning image data.

## 5. Conclusion

Based on the experimental results obtained,

several conclusions can be drawn: First, the Random Forest Method can be used to classify math story problems with an accuracy rate of 84.31%, better than SVM, for story questions that have 1 type of operator. Furthermore, the accuracy is 68.8% for story questions that have 2 or more operators. Second, tree matching is a way that can be used to identify student answers, because it can show students' ways of completing students gradually by seeing if there is a difference between the answer tree and the scanned tree.

## Conflicts of Interest

The authors declare no conflict of interest.

## Author Contributions

Conceptualization, Umi Laili Yuhana; methodology, Ayu Purwarianti and Umi Laili Yuhana; software, Vessa Rizky Oktavia, Ayu Purwarianti; validation, Ayu Purwarianti and Umi Laili Yuhana; formal analysis, Ayu Purwarianti; investigation, Ayu Purwarianti; resources, Vessa Rizky Oktavia, Ayu Purwarianti, Chastine Fatichah and Umi Laili Yuhana; data curation, Vessa Rizky Oktavia, Ayu Purwarianti; writing—original draft preparation Vessa Rizky Oktavia, Ayu Purwarianti; writing—review and editing, Ayu Purwarianti, Chastine Fatichah and Umi Laili Yuhana; visualization, Vessa Rizky Oktavia, Ayu Purwarianti; supervision, Chastine Fatichah, Umi Laili Yuhana; project administration, Ayu Purwarianti and Umi Laili Yuhana; funding acquisition, Ayu Purwarianti and Umi Laili Yuhana.

## Acknowledgments

The first author would like to express my sincere gratitude for Ministry of Education, Culture, Research, and Technology of Republic Indonesia for Postdoc Research Grant.

## References

[1] G. Daroczy, M. Wolska, W. D. Meurers, and H. C. Nuerk, “Word problems: a review of linguistic and numerical factors contributing to their difficulty”, *Front. Psychol.*, Vol. 06, 2015.

[2] R. K. Kedziorski, H. Hajishirzi, A. Sabharwal, O. Etzioni, and S. D. Ang, “Parsing Algebraic Word Problems into Equations”, *TACL*, Vol. 3, pp. 585-597, 2015.

[3] S. Shi, Y. Wang, C. Y. Lin, X. Liu, and Y. Rui, “Automatically Solving Number Word Problems by Semantic Parsing and Reasoning”, In: *Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1132-1142, 2015.

[4] A. Jupri and P. Drijvers, “Student Difficulties in Mathematizing Word Problems in Algebra”, *Eurasia J. Math. Sci. T.*, Vol. 12, No. 9, 2016.

[5] A. A. S. Gunawan, P. R. Mulyono, and W. Budiharto, “Indonesian Question Answering System for Solving Arithmetic Word Problems on Intelligent Humanoid Robot”, *Procedia Computer Science*, Vol. 135, pp. 719-726, 2018.

[6] S. Mandal and S. K. Naskar, “Solving Arithmetic Mathematical Word Problems: A Review and Recent Advancements”, *Information Technology and Applied Mathematics*, Vol. 699, P. Chandra, D. Giri, F. Li, S. Kar, and D. K. Jana, Eds., pp. 95-114, 2019.

[7] A. Prasetya, C. Fatichah, and U. L. Yuhana, “*Parsing struktur semantik soal cerita matematika berbahasa indonesia menggunakan recursive neural network* (Parsing the semantic structure of math story problems in Indonesian using a recursive neural network)”, *Register. Jurnal. Ilm. Teknologi. Sistem. Inf.*, Vol. 5, No. 2, p. 118, 2019.

[8] C. C. Liang, Y. S. Wong, Y. C. Lin, and K. Y. Su, “A Meaning-Based Statistical English Math Word Problem Solver”, In: *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1 (Long Papers), pp. 652-662, 2018.

[9] L. Wang, Y. Wang, D. Cai, D. Zhang, and X. Liu, “Translating a Math Word Problem to a Expression Tree”, In: *Proc. of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1064-1069, 2018.

[10] Q. Liu, W. Guan, S. Li, and D. Kawahara, “Tree-structured Decoding for Solving Math Word Problems”, In: *Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2370-2379, 2019.

[11] A. J. Lado, “Comparison of Neural Network and Random Forest Classifier Performance on Dragon Fruit Disease”, *Wireless Technologies and Intelligent Systems for Better Human Lives*, pp. 351-355, 2021.

[12] N. Z. Fanani, “Two Stages Outlier Removal as Pre-processing Digitizer Data on Fine Motor Skills (FMS) Classification Using Covariance Estimator and Isolation Forest”, *International*

*Journal of Intelligent Engineering and Systems*, Vol. 14, No. 4, pp. 571-582, 2021.

[13] N. Z. Fanani, A. G. Sooai, S. Sumpeno, and M. H. Purnomo, "*Penentuan Kemampuan Motorik Halus Anak dari Proses Menulis Hanacaraka Menggunakan Random Forest* (Determination of Children's Fine Motor Ability from the Hanacaraka Writing Process Using Random Forest)", *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, Vol. 9, No. 2, Art. no. 2, 2020.

[14] A. Bouaziz, C. D. Pallez, C. D. C. Pereira, F. Precioso, and P. Lloret, "Short Text Classification Using Semantic Random Forest", *Data Warehousing and Knowledge Discovery*, Vol. 8646, L. Bellatreche and M. K. Mohania, Eds. *Cham: Springer International Publishing*, pp. 288-299, 2014.

[15] Y. S. Nugroho and N. Emiliyawati, "*Sistem Klasifikasi Variabel Tingkat Penerimaan Konsumen Terhadap Mobil Menggunakan Metode Random Forest* (Variable Classification System of Consumer Acceptance of Cars Using the Random Forest Method)", *Jurnal Teknik Elektro*, Vol. 9, No. 1, Art. no. 1, 2017.