



Article

Synthetic Data Generator for Solving Korean Arithmetic Word Problem

Kangmin Kim  and Chanjun Chun * 

Department of Computer Engineering, Chosun University, Gwangju 61452, Korea

* Correspondence: cjchun@chosun.ac.kr

Abstract: A math word problems (MWP) comprises mathematical logic, numbers, and natural language. To solve these problems, a solver model requires an understanding of language and the ability to reason. Since the 1960s, research on the design of a model that provides automatic solutions for mathematical problems has been continuously conducted, and numerous methods and datasets have been published. However, the published datasets in Korean are insufficient. In this study, we propose a Korean data generator for the first time to address this issue. The proposed data generator comprised problem types and data variations. Moreover, it has 4 problem types and 42 subtypes. The data variation has four categories, which adds robustness to the model. In total, 210,311 pieces of data were used for the experiment, of which 210,000 data points were generated. The training dataset had 150,000 data points. Each validation and test dataset had 30,000 data points. Furthermore, 311 problems were sourced from commercially available books on mathematical problems. We used these problems to evaluate the validity of our data generator on actual math word problems. The experiments confirm that models developed using the proposed data generator can be applied to real data. The proposed generator can be used to solve Korean MWPs in the field of education and the service industry, as well as serve as a basis for future research in this field.

Keywords: math word problem; natural language processing; Korean data generator; Transformer; machine learning

MSC: 68T50; 03B65; 91F20



Citation: Kim, K.; Chun, C. Synthetic Data Generator for Solving Korean Arithmetic Word Problem. *Mathematics* **2022**, *10*, 3525. <https://doi.org/10.3390/math10193525>

Academic Editors: Nebojsa Bacanin and Catalin Stoean

Received: 29 August 2022

Accepted: 22 September 2022

Published: 27 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Advances in deep learning have significantly improved the performance of natural language processing (NLP) in comparison with traditional rules and statistics-based methods. The development of recurrent neural networks (RNNs) initiated the processing of time-series data, such as sentences, via neural structures. In [1], a long short-term memory (LSTM) unit was proposed to alleviate the long-term dependency problem that occurs when the input data is lengthened by adding a memory cell to the RNNs system. In [2], a gated recurrent unit (GRU) was implemented, which simplified the structure of the LSTM and, subsequently, reduced the parameter size while maintaining the performance. The aforementioned studies have established a solid position in the sequence-to-sequence (seq2seq) framework, which comprises an encoder–decoder structure that outputs an input as a sequence in a different domain [3]. Among several studies on sequence modeling, an attention mechanism was proposed to [4,5]. More specifically, the vanilla seq2seq model converts the input into a single context vector of a fixed size during encoding. However, including all the information without omission is challenging. Instead of using only a single fixed-length vector, the attention mechanism additionally employs an attention vector. The attention vector is obtained such that whenever the decoder predicts an output word, it refers to the input associated with that word in the encoder. Owing to the attention vector, each word can acquire more meaningful contextual information. Inspired by the

attention mechanism, the Transformer architecture was proposed in [6], which comprises an encoder–decoder structure designed solely using attention mechanisms, without an RNNs-based network. The architecture exhibited fast learning rates and transformed the models commonly used in machine translation. Recently, large-scale language models [7,8] based on the Transformer structures have emerged and have various applications, such as chatbots [9], translation [10], and text-to-image conversion [11,12].

Designing a math word problems (MWP) solver, where MWPs refer to problems involving mathematical logic, numbers, and natural language, is another field that has been attracting attention owing to advances in artificial intelligence. This task has been continuously addressed since the 1960s [13,14]. When the MWPs solver receives a question as input, it understands the given situation and extracts the necessary information from the sentence. Based on this information, the solver derives a new piece of information: an equation. For this task to be carried out smoothly, the model must have the ability to acquire various domain knowledge, such as that of humans (e.g., the linguistic domain: dozen = 12, and the geometric domain: circle area = radius \times radius \times pi). Moreover, it should also be able to use learned knowledge to understand the context and infer logical expressions. This task primarily requires an understanding of natural language and reasoning skills. Owing to these aspects, the MWPs task has recently been reported to be more suitable than the Turing test for evaluating the intelligence of a model [15].

According to a comprehensive survey on MWPs [16], this task tends to be primarily conducted in English and Chinese, which is also reflected in the published datasets. Meanwhile, research on Korean MWPs remains scarce. The reason for this may be the absence of a Korean dataset. To overcome this problem, a previous Korean study [17] adopted a method for translating datasets containing English MWPs. In this case, word replacement is required during translation. As some words appear in the dataset that reflect cultural differences (e.g., proper nouns and U.S. customary units), this does not fit the Korean context. Additionally, following this process for all data is inefficient. Therefore, in this study, we proposed a data generator instead of translating the datasets. We trained a Korean arithmetic problem solver using machine translation model structures and experimentally evaluated the validity of the data generator by measuring the performance of the solver.

The remainder of this paper is organized as follows. Section 2 provides an overview of the mathematical problem-solving model based on traditional machine learning algorithms and modern approaches. Section 3 presents the problem types and examples, variations that apply to the data, rules for the data, and components of the generated data. Section 4 describes the solver structure and training methods considered in this study. In Section 5, we present the experimental results and discussion. Lastly, Section 6 presents the conclusions of the study and directions for future research.

2. Related Work

The previous research can be broadly divided into three categories:

- A rule-based system is a method to derive an expression by matching the text of the problem to a manually created rule and schema pattern. Ref. [18] used four predefined schemas: change-in, change-out, combine, and compare. The text was transformed into suggested propositions, and the answer was obtained via simple reasoning. Furthermore, Ref. [19] developed a system that can solve multistep arithmetic problems by dividing the schema of [18] in more detail.
- A statistic-based method employs traditional machine learning to identify objects, variables, and operators within the text of a given problem, and the required answer is derived by adopting logical reasoning procedures. Ref. [20] used three classifiers to select problem-solving elements within a sentence. More specifically, the quantity pair classifier extracts the quantity associated with the derivation of an answer. The operator classifier then selects the operator with the highest probability for a given problem from among the basic operators (e.g., {+, −, \times , and /}). The order classifier is used for problems that require an operator (i.e., subtraction and division) related to the order

of the operands. Ref. [21] proposed a logic template called Formula that analyzes text and selects key elements for equation inference to solve multistep math problems. The given problem is identified as the most probable equation using a log-linear model and converted into an arithmetic expression.

These approaches are influenced by predefined data such as annotations for mathematical reasoning; therefore, we cannot obtain satisfactory results if the data is insufficient. Moreover, working with large datasets is time-consuming and expensive.

- Deep learning has attracted increasing attention owing to the activation of big data and the development of hardware and algorithms. The primary advantage of deep learning is that it can effectively learn the features of the large datasets without the need for human intervention. In [22], the Deep Neural Solver was proposed, which introduced deep learning solvers using seq2seq structures and equation templates. Subsequently, this influenced the emergence of various solvers with seq2seq structures [23,24]. Thereafter, a solver with other structures emerged with the advent of the Transformer. Ref. [25] used a Transformer to analyze the notations, namely prefix, infix, and postfix that resulted in enhanced performance when deriving arithmetic expressions. Additionally, Ref. [26] proposed an equation generation model that yielded mathematical expressions from the problem text without equation templates using expression tokens and operand-context pointers.

For more sophisticated MWP solvers, datasets have been proposed besides these methods. Some of the actively used datasets are as follows: Illinois (IL) Data [27] is a dataset composed of 562 items collected from k5learning.com and dadworksheets.com, with one-operation and single-step problems. Problems requiring domain knowledge (e.g., pineapple is a fruit and two weeks comprise 14 days) are removed. The common core (CC) [27] contains 600 data points that were harvested from commoncoresheets.com and comprises multi-operator and multi-step problems. The dataset does not contain irrelevant quantities in the problem text. Math word problems (MAWPS) [28] consists of 2373 one-unknown variable arithmetic problems, and this dataset combines datasets published in previous studies [27,29–31]. Academia sinica diverse mwp dataset-a (ASDiv-A) [32] has 1218 problems with annotations for problem types, grade levels, and equations. Math23 [22] was constructed by crawling several online educational websites. This is a large Chinese dataset comprising 23,161 one-variable linear mathematical word problems. The hybrid math word problems dataset (HMWP) [33] is another Chinese-based dataset that consists of multi-unknown variable problems requiring non-linear equations. The aforementioned datasets comprise English and Chinese, which are not suitable for the Korean MWPs task.

3. Data Generator

This section introduces the Korean math problem generator. An outline of the proposed generator, which has four problem types tailored to the elementary curriculum level, is depicted in Figure 1. Each of these types has subtypes, and the problems are generated for a total of 42 subtypes. The main types of problems and descriptions of the examples are covered in Section 3.1. During the training process, we applied variations to the data to avoid a scenario in which the model derives an answer by simply memorizing the sentence structure of the problem type. Additionally, this enabled the model to handle agglutinative expressions in Korean. Section 3.2 presents these variations. Section 3.3 describes the equation templates used in the previous studies on MWPs and shows the components of the generated math problem data. Finally, Section 3.4 describes the rules for writing Korean math problems.

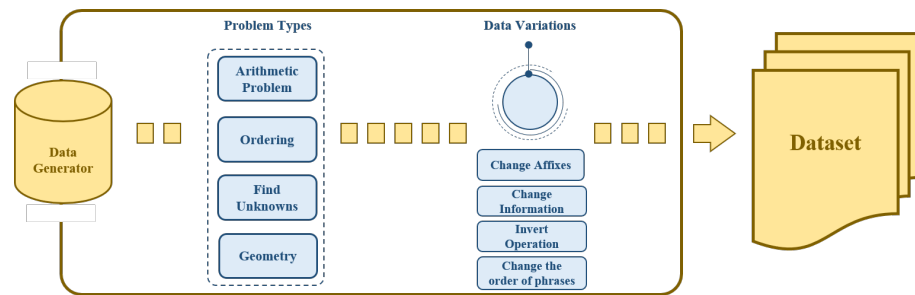


Figure 1. Outline of data generator.

3.1. Types of Problems

Data generators have four types of problems: arithmetic, ordering, finding unknowns, and geometry. The arithmetic problems involve finding an arithmetic expression and the desired answer for a specific situation. Ordering involves answering questions regarding the position or rank of objects in a queue; this is similar to the arithmetic problem. However, this requires an understanding of the sequential situation. Finding unknowns involves identifying a value that satisfies a condition for a given equation filled with unknowns or correcting the result for arithmetic errors and returning the correct result according to the original procedure. Geometry involves finding the area of a figure, perimeter, or length of a side for a given a geometric figure. Other studies [34–36] presented geometric figures as images. However, as the focus of our study was on the linguistic abilities of the model, we only used sentences.

Tables 1–4 presents examples of problems. The number in parentheses indicates the number of subtypes of a given type. A total of 42 subtype problems are present.

Table 1. Examples of the arithmetic problem type.

Types of Problems		Descriptions/Example of Subtype Problems
	Description	Finding an arithmetic expression and the desired answer for a specific situation
Arithmetic Problem (12)	Example 1	<p>English I was going to distribute the yuzu evenly to 59 people, but I accidentally gave it to 80 people. I gave 5 to each person and there were 43 left. If this yuzu is evenly distributed among 59 people, how many will be the maximum per person?</p> <p>Korean 유자를 59명에게 똑같이 나누어 주어야 할 것을 잘못하여 80명에게 똑같이 나누어 주었더니, 한 사람당 5씩 주고 43개가 남았다. 이 유자를 59명에게 똑같이 나누어 주면 한 사람당 최대한 몇 개씩 가지게 되는지 구하시오.</p> <p>Equation $((80 \times 5) + 43)/59$</p>
	Example 2	<p>English Pil-Gyu gave 84 persimmons to Seong-Won and 22 to Yoon-Sang. Of the persimmons he had, 37 were left. Find out how many persimmons Pil-Gyu had at the beginning.</p> <p>Korean 필규는 감을 성원에게 84 개를 주고 윤상에게 22 개를 주었더니 가지고 있던 감중에서 37 개가 남았다. 처음에는 필규가 가지고 있던 감은 몇 개인지 구하여라.</p> <p>Equation $84 + 22 + 37$</p>
	Example 3	<p>English Find the sum of the odd numbers. The range is from 1 to 11.</p> <p>Korean 홀수의 총합을 구하시오. 범위는 1부터 11까지이다.</p> <p>Equation $1 + 3 + 5 + 7 + 9 + 11$</p>
	Example 4	<p>English The Japanese scores of Song-Woo, Dae-Yong, and Jong-Woo are 35, 70, and 54, respectively. Except for these three students, the average Japanese score was 54. If Song-Woo’s class has 81 students, what is the average Japanese score for the class?</p> <p>Korean 송우, 대용이, 종우의 일본어 점수는 각각 35점, 70점, 54점이다. 이 3명을 제외한 나머지 학급의 일본어 점수 평균은 54점이다. 송우네 학급 인원수가 81명일 때, 학급 일본어 평균 점수는 몇 점인지 구하시오.</p> <p>Equation $(54 \times (81 - 3) + 35 + 70 + 54)/81$</p>

Table 2. Examples of the ordering type.

Types of Problems		Descriptions/Example of Subtype Problems	
Ordering (7)	Description	Finding the correct answer required in an ordered situation.	
	Example 1	English	47 male students are sitting in a row. If there are 32 boys behind Su-Jeong, how many boys are sitting in front of him?
		Korean	47명의 남학생들이 한 줄로 줄을 앉아 있다. 수정이의 뒤에 32명의 남학생 있다면, 수정이의 앞에 앉아 있는 남학생은 몇 명이 있을까?
		Equation	$(47 - 1) - 32$
	Example 2	English	In the midterm exam, Yong-Hwan is placed 42nd and Doo-Hyeon is placed 40th. If Seong-Jae is ranked higher than Yong-Hwan and lower than Doo-Hyeon, what is Sung-Jae's rank?
		Korean	중간고사 시험에서 용환이는 42위를 하였고, 두현이는 40위를 기록했다. 성재는 용환이보다 순위가 높고, 두현이보다는 순위가 낮다고 한다면 성재의 순위는?
		Equation	$(40 + 42)/2$
	Example 3	English	Twenty-four people are standing in a line in order from the tallest customer. Min-Jeong is standing 22nd from the front. If she lines up again in order of the shortest person, where is she standing?
		Korean	키가 큰 손님부터 순서대로 24명이 한 줄로 서 있습니다. 민정이가 앞에서부터 스물 두 번째에 서 있습니다. 키가 작은 사람부터 순서대로 다시 줄을 서면 민정이는 몇 번째에 서 있습니까?
Equation		$24 - 22 + 1$	

Table 3. Examples of the finding unknowns type.

Types of Problems		Descriptions/Example of Subtype Problems	
Finding Unknowns (11)	Description	(a) Finding the unknowns that satisfy the condition of an expression. (b) Finding the result from the correct operation given the situation, in which the operation is wrong.	
	Example 1 (a)	English	Find the natural number corresponding to A in the addition expression of three-digit natural numbers 'A5C + 1B5 = 960'
		Korean	세 자리 자연수의 덧셈식 'A5C + 1B5 = 960'에서 A에 해당하는 자연수를 구하여라.
		Equation	$(960//100)-(100//100)-(10-6+5+(10-((960\%10)\%10)+5-1)//10-1)//10$
	Example 2 (a)	English	A and B are natural three-digit numbers. 996 is 3 less than A and B is 92 less than 182. Find the sum of A and B.
		Korean	A, B는 자릿수가 세 개인 자연수이다. A보다 3 작은 수는 996이고, B는 182보다 92 작은 수이다. A와 B의 합을 구하여라.
		Equation	$(996 + 3) + (182 + 92)$
	Example 1 (b)	English	You get 90 when you subtract 19 from an unknown number. What is the result of subtracting 29 from the unknown number?
		Korean	어떤 수에서 19를 뺀 때 90가 되었습니다. 어떤 수에서 29를 빼면 얼마가 되는지 구하시오.
		Equation	$90 + 19 - 29$
	Example 2 (b)	English	If you multiply an unknown natural number by 18, subtract 30, add 16, and divide by 1 you get 526. What is the unknown natural number?
Korean		어떤 자연수에 18을 곱하고 나서 30을 빼고, 16을 더한 값을 1로 나눈다면 526이 된다고 합니다. 어떤 자연수를 구하시오.	
Equation		$((526 \times 1) - 16) + 30 / 18$	

Table 4. Examples of the geometry type.

Types of Problems	Descriptions/Example of Subtype Problems		
Geometry (12)	Description	Finding the area, perimeter, or length of a side for a given geometric figure.	
	Example 1	English	If you have a circle of radius 21, what is the area of the circle?
		Korean	반지름의 길이가 21인 원이 있다면, 원의 넓이는 얼마 인지 구하시오.
		Equation	$\pi \times 21 \times 21$
	Example 2	English	If you have a square with a side length of 40 m, how many square meters is it?
		Korean	한 변의 길이가 40m인 정사각형이 있다면, 정사각형의 넓이는 몇 m ² 입니까?
		Equation	40×40
	Example 3	English	In math class, Jeongseok created a rectangle with yarn. There was no leftover yarn and he did not run out of thread. The total length of the yarn was 176 kilometers. If a rectangle is 58 kilometers wide, what is the vertical length?
		Korean	수학시간에 정석이는 실로 직사각형을 만들었습니다. 사용한 실은 남지도 모자라지도 않았습니다. 실은 총 176 킬로미터 이고, 직사각형의 가로 길이는 58 킬로미터 라면, 세로 길이는 몇 킬로미터입니까?
		Equation	$(176 - (58 \times 2))/2$
	Example 4	English	Na-Rae drew a rectangle with a perimeter of 80 km . If the width of the rectangle is three times the length, how many kilometers is the width?
		Korean	나래는 둘레가 80km인 정사각형을 그렸습니다. 이 정사각형의 가로 길이가 세로 길이의 3배라고 하면, 가로 길이는 몇 km입니까?
Equation		$((80/2)/(3 + 1)) \times 3$	

3.2. Data Variations

In MWP, questions are presented in different forms even for the same problem type. As Korean has agglutinative characteristics, providing additional linguistic expressions is possible by changing the suffix of the root in the problem text. An agglutinative language is a form of language characterized by the addition of prefixes, suffixes, and other morphemes to the roots to form words. To develop a robust MWP solver, the model must learn as many sentences as possible. Therefore, we apply variations when generating problems, such that the model can capture many phrases. The variation method can be classified into four types: alternate affixes, change information, replace operator, and change the order of a phrase. The alternate affixes type changes the affixes of the root, such that the model learns various phrasing. The change information type converts a proper noun or object, which is the information provided in the problem. The replace operator type transforms a word provided for operator inference with another operator. These three variations guide the model to achieve proper reasoning without being constrained to the specific information of a given question. The change in the order of phrase type swaps the order of the sentences. This prevents the model from merely memorizing the structure of the problem superficially. In addition to these techniques, we applied the random.seed function from Python for the generator to prevent operands and variants from appearing consistently. When generating a dataset, the random numbers generated from each uniquely assigned seed value select the variations. This enables each dataset to receive less consistent data. Examples of the data variations are listed in Table 5.

Table 5. Types and examples of variations. Colored text represents changes via variation.

Variation		Examples
Alternate Affixes	Original	English Hyun-Woo collected 1 and 40, and Han-Gyul collected 25 and 59. Which child has a larger total?
		Korean 현우는 1와 40를 모았고 , 한결은 25와 59를 모은 상태입니다. 어떤 아이의 총 수가 훨씬 클까요?
	Variation	English Hyun-Woo is collecting 1s and 40s. Han-Gyul gathered 25 and 59. Which child has a larger total?
		Korean 현우는 1와 40를 모으고 있고 . 한결은 25와 59를 모았습니다 . 어떤 아이의 총 수가 훨씬 클까요?
Change Information	Original	English Shin-Yeol picked two sequential numbers. When the sum of the two numbers drawn is 171, what is the greater of the two numbers Shin-Yeol picked?
		Korean 신열 이는 연속된 두 수를 뽑았다. 뽑은 두 수의 합이 171이었을때, 신열 이가 뽑은 두 수 중 큰 수는 몇인지 구하시오.
	Variation	English Joo-Hyung picked two sequential numbers. When the sum of the two numbers drawn is 171, what is the greater of the two numbers picked by Joo-Hyung ?"
		Korean 주형 이는 연속된 두 수를 뽑았다. 뽑은 두 수의 합이 171이었을때, 주형 이가 뽑은 두 수 중 큰 수는 몇인지 구하시오."
Replace Operator	Original	English I subtracted 188 from an unknown number to get 833. Guess what value would be obtained if 248 was added to the unknown number.
		Korean 모르는 수에서 188를 뺐더니 833가 되었다. 그렇다면 모르는 수에서 248를 더했을 경우 어떤 값이 나올지 맞추어라.
	Variation	English If you add 188 to an unknown number, you get 833. Guess what value will be obtained if 248 is subtracted from the unknown number.
		Korean 모르는 수에서 188를 더했더니 833가 되었다. 그렇다면 모르는 수에서 248를 뺐때 얼마가 될지 맞추어라.
Change the order of a phrases	Original	English When I is divided by 129 , the quotient is J, and the remainder is K. When I, J, and K are natural numbers , the quotient and the remainder are the same. Find the largest divisor of J.
		Korean I를 129로 나누었을 때 몫은 J이며, 나머지는 K가 됩니다. I, J, K는 자연수일 때 , 몫과 나머지는 같습니다. 나누어지는 수 J 중 가장 큰 수를 구하시오.
	Variation	English I, J, and K are natural numbers. If you divide I by 129 , the quotient is J and the remainder is K. Moreover, the quotient and the remainder are equal. Find the largest divisor of J.
		Korean I, J, K는 자연수일 때, I를 129로 나누면 몫은 J이고, 나머지는 K가 됩니다. 또한, 식에서 몫과 나머지는 같습니다. 나누어지는 수 J 중 가장 큰 수를 구하시오.

3.3. Components of the Generated Data

The main aspect of the previous studies was the method using an equation template. This template is an abstract form of an equation when the types and positions of the operators are the same, and only the operands are different depending on the question. More specifically, if the equation expressed by the problem is $X = 2 + 3$, a generalized form such as $X = n_1 + n_2$ is indicated. In this approach, a template suitable for the problem is first determined via classification to solve a given mathematical problem [37]. Sequentially, numerical information is extracted from the text within the problem and the template is populated. This enables the model to easily draw mathematical inferences. However, the lack of a predefined template has the disadvantage of low prediction accuracy. On the other hand, Ref. [26] proposed a template-independent equation generation model that is trained to generate equations by extracting equation-related information from the problem text. We leveraged this approach not to consider the template sparsity issue. In conclusion, our generated data comprises questions, equations, and answers.

3.4. Rule of Data

The problem text consists of the Korean language, alphabets, Arabic numerals, the SI international system of units, and punctuation marks. In particular, punctuation marks only include those mentioned in the Hangeul Spelling Appendix. If uppercase letters of the alphabet are listed without punctuation marks or spaces, this indicates each digit (i.e., Table 3, Finding Unknowns Example 1). Words expressing different units, such as km, m, cm, and kg are included in the question text and can be expressed in lowercase English or Korean (i.e., Table 4, Geometry Examples 1 and 2). This is one of the methods for the model to learn various linguistic expressions. The level of words and difficulty of the problems are limited to the Korean elementary curriculum. The equation has the basic operators $\{+, -, /, \times\}$ and Python operators $\{//, \%\}$. Only one correct answer exists for each question. Even if units and names are mentioned in the question, only numbers are specified in the correct answers, which are written as positive rational numbers. When a decimal answer is required, it is rounded to three decimal places.

4. Materials and Methods

In this section, we describe the solver structure and training methods. We consider that extracting keywords from a given problem text to generate mathematical formulae is equivalent to translating from natural language to mathematical formulae. Therefore, we adopted the model used for machine translation as the structure of the solver. Section 4.1 describes the machine translation framework that we adopted. Section 4.2 provides the hyperparameters, optimizer, and the number of epochs applied to these models. Section 4.3 presents a word embedding method that was applied to the solver to improve its ability to understand the meanings of words and improve accuracy.

4.1. Architecture

4.1.1. Vanilla Seq2seq

Seq2seq is a basic model in machine translation and comprises an encoder and a decoder module. The encoder compresses the information in the input sequence into a vector called a context vector. The decoder takes a context vector as initial hidden states and generates a sequential output. The encoder and decoder consist of an RNNs-based system. In our experiments, we used GRU cells and not vanilla RNNs.

4.1.2. Seq2Seq with Attention Mechanism

Vanilla Seq2seq has two drawbacks. The first is gradient vanishing, which is a chronic problem of RNN systems. The second is the loss of input sequence information caused by compressing all information into a fixed-size vector. These result in a decrease in prediction accuracy as the input sentence increases in length. The attention mechanism is a method for tackling this issue. The basic idea is that for each time step the decoder predicts the output word, it once again consults the entire input sentence from the encoder. However, instead of referring to the entire input sentence at the same rate, the decoder pays close attention to the portion of the input word that is connected to the word that will be predicted at that instant. The structure of the seq2seq with the attention model that we considered in our experiment is similar to the aforementioned vanilla structure. However, the difference is that the attention mechanism is added to the decoder. Figure 2 illustrates this structure. The attention equation is expressed as follows [5]:

$$\text{score}(h_t, \bar{h}_s) = h_t^T W_a \bar{h}_s, \quad (1)$$

$$a_t(s) = \frac{\exp(\text{score}(h_t, \bar{h}_s))}{\sum_{s'} \exp(\text{score}(h_t, \bar{h}_{s'}))}, \quad (2)$$

$$c_t = \sum_n a_t(n) \bar{h}_n, \quad (3)$$

$$\tilde{h}_t = \frac{\exp(W_b[c_t; h_t])}{\sum_{s'} \exp(W_b[c_{s'}; h_{s'}])} \tag{4}$$

where *score* represents the attention score function that calculates the similarity for two given vectors, h_t represents the hidden state of the decoder at each time step t , \tilde{h}_s denotes the hidden state of the encoder at each word s . W_a, W_b represent the model parameters, which are trainable weights. Moreover, a_t represents the attention weight, which considers the softmax from all scores obtained at time t . c_t is the encoder context vector for time t , which is obtained by the weighted summing of the attention weights and the words information of the encoder, and \tilde{h}_t represents the predicted word obtained via the context vector and the hidden state of the decoder.

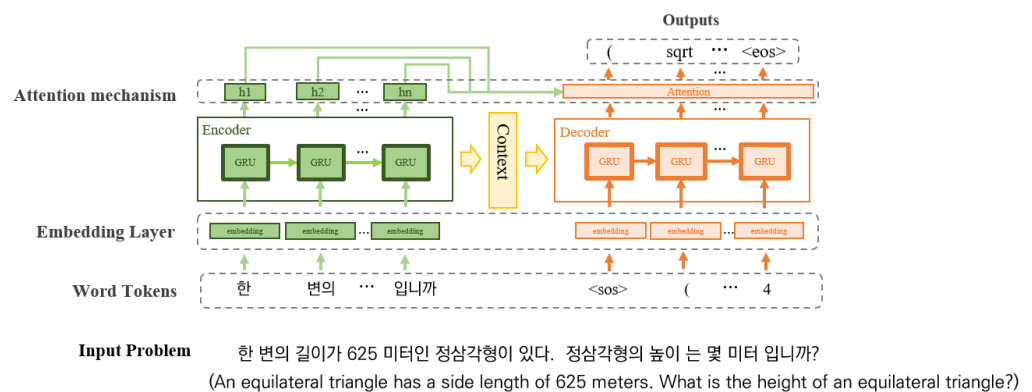


Figure 2. Structure of the seq2seq with the attention mechanism. The green square h represents the hidden state of each word acquired via the encoder.

4.1.3. Transformer

The attention mechanism is used to alleviate the long-term dependence problem; however, the problem of sequential nature remains. The Transformer is a method of constructing an encoder–decoder structure by only using attention mechanisms by removing the RNNs system from the existing structure. The structure is depicted in Figure 3. The Transformer does not receive and process data sequentially but receives a sequence at a time and processes it using attention. This approach accelerates the computation of the model, as it is relatively easy to train and parallelize.

Representative technologies used by a Transformer to replace RNNs are as follows: self-attention, multi-head attention, and positional encoding. Self-attention is a method of calculating the relevance of words appearing in an input sentence and reflecting these on the network. This technology generates query, key, and value vectors from the vectors of each word and subsequently uses these to calculate the attention score of each word. The advantage of self-attention is that it can calculate the association between input words and is not affected by the long-term dependency problem because it can connect the words directly without sequential processing. Multi-head attention is a method of dividing self-attention into parallel head units. Each head calculates a self-attention for the word. Then, concatenates the values obtained from the heads to represent the word. It serves to reflect the various information in the word. Positional encoding is a technology for reflecting the positional information of each word. A Transformer does not receive sequences sequentially; therefore, the context information indicated by the word order cannot be grasped. To avoid this issue, a Transformer gets the sequence information via a periodic function and adds it to the word vector to figure out the word order.

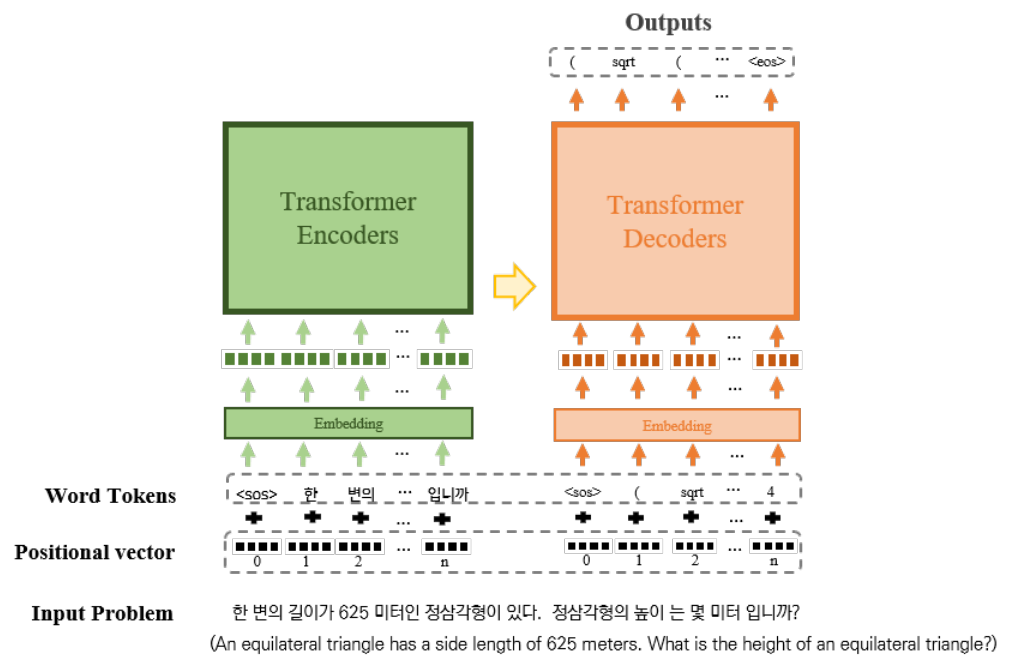


Figure 3. Structure of the Transformer model. The encoder of the Transformer delivers the semantic information of the input sentence obtained through self-attention and multi-head attention to the decoder.

4.2. Hyperparameters of Models

Table 6 lists the hyperparameters and values that each model should consider. Furthermore, Adam was used as the optimizer, and early stopping, which stops training if the validation loss does not decrease for 20 epochs, was employed to prevent overfitting.

Table 6. Hyperparameters and values to consider for each model. Scratch indicates that the model is trained from scratch without an embedding algorithm.

Hyperparameters	Seq2seq	Seq2seq (with Attention)	Transformer			
Word Embedding	FastText	FastText	FastText	GloVe	SGNS	Scratch
Embedding Size	[256]	[256]		[256, 384]		
Hidden Size	[256]	[256]		[256, 384]		
Number of Layers	[3, 4]	[3, 4]		[3, 4]		
Learning Rate	$[1 \times 10^{-4}, 5 \times 10^{-4}]$	$[1 \times 10^{-4}, 5 \times 10^{-4}]$	$[1 \times 10^{-4}, 5 \times 10^{-4}, 5 \times 10^{-5}]$			
Dropout	[0.1, 0.2, 0.4]	[0.1, 0.2, 0.4]		[0.1, 0.2, 0.4]		
Batch Size	[1024]	[1024]		[256]		
FFN Size	-	-		[512, 768]		
Head	-	-		[4, 8]		
# of Params	5.8 M	7.1 M	7.4 M			
Epochs		300				

4.3. Word Embedding

In NLP, text must be properly converted to numbers such that the computer can understand it. Research to efficiently capture the meaning of words is being actively conducted because the performance of downstream operations depends on the expression of words. We used the representative word-embedding algorithm that has been studied thus far to capture the meaning of the words in the problem text. The three algorithms adopted are as follows:

- Skip-gram with negative sampling (SGNS) is a method of predicting the context word, which is the surrounding word of the target word. Vanilla skip-gram is very inefficient

because it updates all word vectors during backpropagation. Therefore, negative sampling was proposed as a method to increase computational efficiency. First, this method randomly selects words to generate a subword set that is substantially smaller than the entire word set. Subsequently, this method performs positive and negative binary classification of whether the subset words are near the target word. This method updates only the word vectors that belong to the subset. In this manner, SGNS can perform vector computation efficiently. [38].

- Global vectors for word representation (GloVe) is a method to compensate for the shortcomings of Word2Vec and latent semantic analysis (LSA). Because LSA is a count-based method, comprehensive statistical information can be obtained about words that appear together with a specific word. However, the performance of LSA is poor in the analogy task. By contrast, Word2Vec outperforms LSA in this task but cannot reflect statistical information because Word2Vec can only see context words. The GloVe is a method for using both embedding mechanisms [39].
- FastText uses an embedding learning mechanism identical to that of Word2Vec. However, Word2Vec treats words as indivisible units, whereas FastText treats each word as the sum of character unit n-grams (e.g., tri-gram, apple = app, ppl, ple). Owing to this characteristic, FastText has the advantage of being able to estimate the embedding of a word even if out-of-vocabulary problems or typos are present [40].

We trained word vectors with the generated training dataset because pretrained Korean word embeddings in identical situations have not been published. The representation of the trained word vector spanned 256 dimensions, and the embedding vector was applied to the embedding layer of the models.

5. Result and Discussion

We adopted accuracy and the bilingual evaluation understudy (BLEU) score as the evaluation metrics. Accuracy represents the percentage of correct answers to the predicted equation. In this case, the predicted expressions indicate those that have successfully navigated the exception handling process of Python without producing a syntax error. The BLEU score measures the similarity between the predicted and correct equations. The BLEU score is computed according to the following [41]:

$$p_n = \frac{\sum_{s \in c} \min(C(s, c), C(s, r))}{\sum_{s \in c} C(s, c)}, \quad (5)$$

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-\frac{c}{r})} & \text{if } c \leq r \end{cases} \quad (6)$$

$$BLEU = BP \times \exp\left(\sum_{n=1}^N w_n \log p_n\right), \quad (7)$$

where p_n represents the modified n-gram precision, C indicates the number of n-gram word s in a given sentence, c denotes the predicted sentence and r denotes the correct sentence, N indicates the maximum length of the n-gram, which was set to four. Moreover, w_n denotes the weight applied to each n-gram. In this experiment, the weight of 0.25 was applied. Additionally, BP represents the brevity penalty, and its value is determined by the lengths of c and r .

A total of 210,000 data points were generated, and 150,000 data points constituted the training set. The validation and test datasets contained 30,000 data points each.

Figure 4 illustrates the losses (left) and performances (right) of the model for the validation dataset. Green indicates the seq2seq model, blue indicates seq2seq with attention, and red indicates the Transformer. In the plot on the left, the Transformer model converges at 46 epochs and displays the fastest training speed by stopping early at 66 epochs, whereas the seq2seq structure-based models completed all 300 epochs. The loss in the seq2seq model with attention tended to be higher than that in the vanilla seq2seq model. However, the loss

was not a definite indicator of performance. The performances of each model are shown in the plot on the right. The solid lines represent the accuracy, and the dashed lines represent the BLEU scores. We confirmed that the BLEU score and accuracy of the seq2seq with attention were higher than those of the vanilla seq2seq model. This demonstrates that the attention mechanism focuses on keywords to generate equations from sentences. Moreover, we affirmed that the Transformer achieves an accuracy similar to that of the model with attention applied even with a reduced number of epochs.

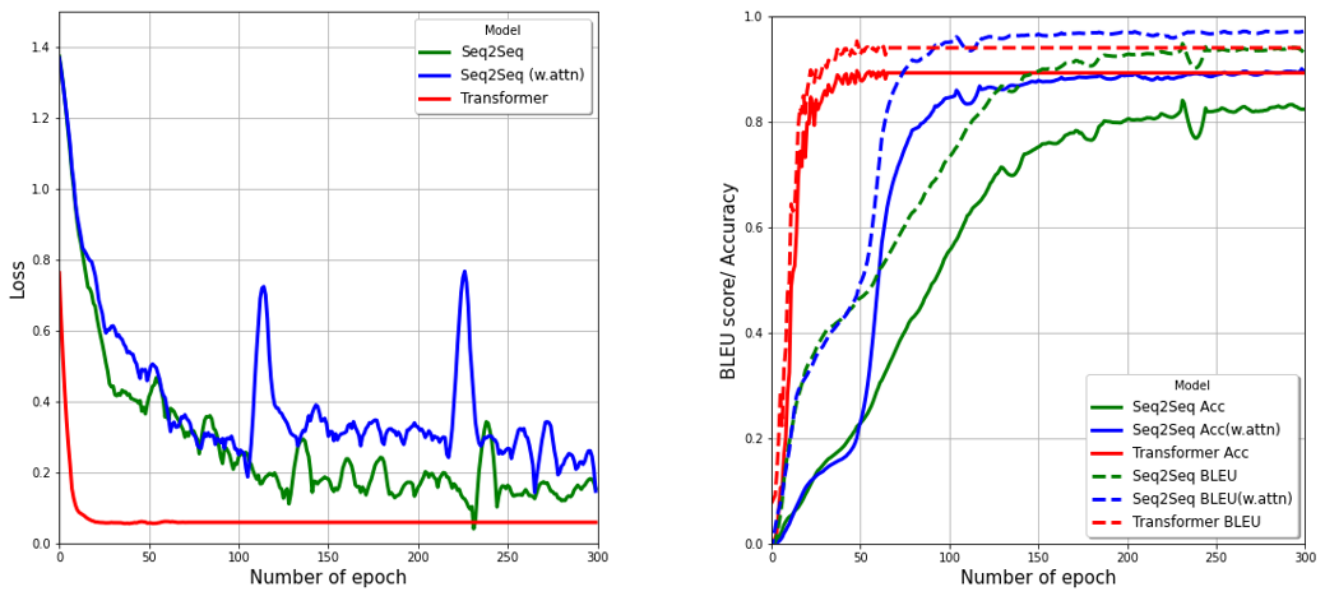


Figure 4. (Left) Losses of the models on the validation dataset. (Right) Accuracy and BLEU scores obtained from the validation dataset of the model.

Table 7 presents quantitative performance data of the models and embedding method for the validation and test datasets. Between the two datasets, the performances of the models are almost the same. The Transformer from scratch model showed the largest performance difference, although this model only differed by 0.49% in BLEU score and 0.29% in accuracy. We assigned each dataset with a different random seed value, which suggested that the model faces different variations for each dataset. Nevertheless, these results demonstrated that the model is robust to variations. More specifically, our data generator enables the model to concentrate on critical terms for equation derivation rather than only surface data.

Table 7. Quantitative performance results obtained using the validation and test datasets for each model and embedding algorithm.

Model	Embedding	Validation		Test	
		BLEU Score (%)	Accuracy (%)	BLEU Score (%)	Accuracy (%)
Seq2seq	FastText	93.54	82.11	93.33	82.06
Seq2seq (with attention)	FastText	97.15	89.56	97.04	89.39
Transformer	Scratch	85.66	79.92	85.17	79.63
	FastText	93.61	89.05	93.33	89.08
	SGNS	94.94	90.27	94.76	90.38
	GloVe	94.71	90.93	94.75	90.97

The attention model with FastText embedding achieved the highest BLEU score of over 97%, and the Transformer with GloVe recorded achieved 90.9% accuracy. The Transformer is more accurate than the attention applied seq2seq; however, the BLEU score is lower than the seq2seq with attention. Table 8 lists the reasons for this finding.

Three equation forms are produced by the Transformer: predict the correct equation, change the order, and add parentheses. The predict the correct equation forms refers to the situation in which the anticipated and actual equations are identical. Changing the order indicates a prediction result obtained by changing the order of the numbers. Adding parentheses represents a case of solving by adding parentheses that are not in the correct equation. The order is changed and parentheses are added frequently in the Transformer. The equations are considered identical in both cases for the human domain; however, the BLEU score reflecting the n-gram is not, which appears to be the cause of the score gap.

Table 8. Equation form that manifests when a Transformer anticipates an equation.

Type of Predicted Equation	Input Problem	Correct Equation	Predicted Equation
	There are 25 apples in a box. How many apples are in all 5 boxes?	25×5	25×5
Predict the Correct Equation	There is a trapezoid with an upper side length of 25 cm, a lower side length of 34 cm, and a height of 12 cm. What is the area of the trapezoid?	$(25 + 34) \times 12/2$	$(25 + 34) \times 12/2$
	There were 27 apples in the box. Five of them were discarded. If you put 18 more apples in this box, how many apples are in the box	$27 - 5 + 18$	$27 - 5 + 18$
Change the Order	1 dozen pencils are 12. How many are 6 dozen pencils in total?	12×6	6×12
	What is the perimeter of a parallelogram, the side of which is 18 cm long and the other side is 12 cm long?	$(18 + 12) \times 2$	$2 \times (18 + 12)$
	There are machines that manufacture 900 toys per day. How many toys can the machine manufacture in 7 days without a break?	900×7	7×900
Add parentheses	What is the area of a triangle with a base of 13 m and a height of 8 m?	$13 \times 8/2$	$(13 \times 8)/2$
	I have a rectangular notebook with a perimeter of 46 cm and a height of 9 cm. How many cm is the width of this notebook?	$(46 - 9 \times 2)/2$	$(46 - (9 \times 2))/2$
	The sum of the four sides of a rectangle is 32 cm. If the width of this rectangle is 7 cm, how many cm is the length?	$(32 - 7 \times 2)/2$	$(32 - (7 \times 2))/2$

We conducted additional experiments to verify that our data generator is valid in real mathematical problems. The actual dataset is composed of 311 problems harvested from a commercially available book on problems. It includes both types that were utilized in training and those that were not, for which two reasons exist. The first is to confirm that the model is overfitting the generated dataset and evaluate the performance of the solver on real problems. The second is to evaluate the closeness of the predictions of the model and those of the correct equation, even when the type is unknown. Table 9 presents the performance of the models on a real-world dataset. In contrast to Table 7, the best performance was obtained when FastText was used as the embedding layer. The Transformer with FastText performed the best by scoring 34.71% and 22.32% in the accuracy and BLEU scores, respectively. When this embedding was applied to the seq2seq-based models, the vanilla seq2seq model achieved an accuracy of 16.39% and a BLEU score of 13.52%. In addition, the seq2seq model with attention achieved 20.52% accuracy and 17.02% BLEU score. Compared with the validation results, the performance drop of the seq2seq-based model is more significant than that of the Transformer. These findings suggest that the seq2seq-based model tended to overfit the training data. The outcome of the Transformer also demonstrates that our data generator is valid for solving some of the actual math problems. Table 10 lists several samples with success and failure of the Transformer using FastText.

Table 9. Outcomes of the model when applied to real-world data.

Actual Math Word Problem Dataset			
Model	Embedding	BLEU Score(%)	Accuracy(%)
Seq2seq	FastText	13.52	16.39
Seq2seq (with attention)	FastText	17.02	20.57
Transformer	Scratch	11.22	15.11
	GloVe	15.94	24.43
	SGNS	19.99	27.00
	FastText	23.23	34.72

Table 10. Examples of correct and incorrect equations from the Transformer with FastText, which perform the best on the real-world problem dataset.

	Input Problem	Correct Equation	Predicted Equation
English	There are 268 fewer olives than quince, and there are 368 olives and quince in total. Find out how many quinces are there.	$((368 - 268)/2)+268$	$((368 - 268)/2)+268$
Korean	올리브가 모과보다 268개 더 적고 올리브와 모과가 모두 합해서 368개 있다. 모과는 몇 개인지 구하여라.		
English	We need to subtract 475 from an unknown number A, but the result of subtracting 214 by mistake is 911. What is the result of the original calculation?	$911 + 214 - 475$	$911 + 214 - 475$
Korean	미지수 A에서 475를 빼야 하는데 실수로 214를 뺀 결과, 911가 나오게 되었다. 이때 원래대로 계산한 결과는 무엇일까요?		
English	One box can hold 34 apples. How many apples can 4 boxes hold?	34×4	34×4
Korean	사과를 한 상자에 34개씩 담을 수 있습니다. 사과 4상자에는 사과를 모두 몇 개 담을 수 있습니까?		
English	A total of 808 friends are sitting in a row. 268 people are sitting in front of Chae-Yeon. How many friends are sitting behind her?	$(808 - 1) - 268$	$(808 - 1) - 268$
Korean	808명의 친구들과 한 줄로 앉아 있습니다. 채연이의 앞에는 268명이 앉아 있다고 합니다. 채연이의 뒤에는 몇 명의 친구들이 앉아있을까요?		
English	There are 4 comic books, 2 novels, and 2 textbooks on the bookshelf. How many books are on the bookshelf?	$4 + 2 + 2$	4×2
Korean	책꽂이에 만화책 4권, 소설책 2권, 교과서 2권이 꽂혀 있습니다. 책꽂이에 꽂혀 있는 책은 모두 몇 권 인지 구하세요		
English	A triangle has a base of 17 cm and a height of 16 cm. What is the area of the triangle?	$17 \times 16/2$	$(17 \times 16)/2$
Korean	밑변이 17 cm이고 높이가 16 cm인 삼각형이 있습니다. 이 삼각형의 넓이는 몇 cm ² 입니까?		
English	I have a rectangle with a width of 22 cm and a perimeter of 64 cm. What is the length of the rectangle in cm?	$(64 - 22 \times 2)/2$	$(22 - (64 \times 2))/2$
Korean	가로 길이가 22 cm이고 둘레는 64 cm인 직사각형이 있습니다. 이 직사각형의 세로는 몇 cm입니까?		
English	I am trying to climb to the observatory of a certain building. If I have to climb 249 m and have reached 153 m, find out how much more I need to climb.	$249 - 153$	$153 + 249 - 153$
Korean	어떤 건물의 전망대에 올라가려고 합니다. 전망대까지 249 m를 올라가야 하는데 153 m까지 올라왔다면 몇 m를 더 올라가야 할지 구하세요.		

The examples up to the fourth row were from the test set. To the model, these were familiar types; therefore, a high concordance rate was observed. The following two rows were cases of accurate responses to real-world questions. Even though some discrepancies were present between the expressions, the model provided answers through knowledge learned from the generated data. The last two rows were cases of errors found in the actual dataset. Even in these cases, the solver was able to find the required values to derive the answer within the text. However, this agent had limitations because it did not learn all

the types of real-world problems and the contextual meaning of words, which eventually resulted in incorrect arithmetic expressions.

6. Conclusions

This study presented a novel Korean data generator for MWP tasks. The proposed generators consist of problem types and data variations. We primarily dealt with arithmetic problems, ordering, finding unknowns, and geometry. These had subtypes, which were the syntactic expressions of the problem that may exist in each category, and a total of 42 subtypes were prepared. Variations were applied to the subtype problems when generating the dataset. Two reasons exist for the data variations. The first is to enable the model to understand the meaning of the root without being confused by the agglutination of Korean. The second is to prevent the model from deriving the answer through the extrinsic structure of the sentence.

We created a total of 210,000 data points via the proposed data generator. The training dataset was composed of 150,000 data points, and the validation and test datasets were allocated 30,000 data points each. As the problem solver structures, seq2seq, seq2seq with attention, and the Transformer were employed, which are often used in the machine translation domain. The FastText, GloVe, and SGNS word-embedding algorithms were employed to enable the models to sufficiently understand the meaning of words.

The experimental results confirmed that the Transformer structure solver recorded an accuracy of 90.97% on the generated dataset. We performed further experiments to confirm that the trained model could solve the actual data. The real dataset comprises 311 pieces of data harvested from a commercially available book on mathematical problems, which also included non-learned types for objective evaluation of the model. In the final result, the Transformer with FastText recorded an accuracy of 34.72%. This performance demonstrated that the trained model can respond to the generated data and real-world problems.

This study can provide training data for models that automatically solve Korean math problems in fields such as education and the service industry. In addition, this data can be used for Korean language learning and evaluating intellectual ability for future multilingual language models.

Future Research Directions

We confirmed that the proposed data generator partially reflects real-world data. However, we cannot overlook the difference between the generated and the actual data. In future research, additional problem types and variations are needed to make the model robust to exceptional data that do not belong to the four categories described. Furthermore, by using a large-scale language model that adopts the Transformer structure, an approach that captures rich word expressions and improves the accuracy of equations is required.

The research on Korean data generation for MWPs is still in its early stages; however, we expect that this study will serve as a basis for future research.

Author Contributions: Writing—original draft preparation, methodology, and experimental setup, K.K. and supervision and project administration, C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the research fund from Chosun University, 2021.

Data Availability Statement: Not applicable

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780.
2. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. In Proceedings of the NIPS 2014 Workshop on Deep Learning, Montreal, ON, Canada, 12 December 2014.
3. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*; MIT Press: Cambridge, MA, USA, 2014; Volume 27, pp. 3104–3112.

4. Bahdanau, D.; Cho, K.H.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
5. Luong, M.T.; Pham, H.; Manning, C.D. Effective Approaches to Attention-based Neural Machine Translation. In Proceedings of the Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1412–1421.
6. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*; MIT Press: Cambridge, MA, USA, 2017; Volume 30.
7. Kenton, J.D.M.W.C.; Toutanova, L.K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the NAACL-HLT, Los Angeles, CA, USA, 2–7 June 2019; pp. 4171–4186.
8. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*; MIT Press: Cambridge, MA, USA, 2020; Volume 33, pp. 1877–1901.
9. Caldarini, G.; Jaf, S.; McGarry, K. A literature survey of recent advances in chatbots. *Information* **2022**, *13*, 41.
10. Dabre, R.; Chu, C.; Kunchukuttan, A. A survey of multilingual neural machine translation. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–38.
11. Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; Sutskever, I. Zero-shot text-to-image generation. *Proc. Mach. Learn. Res.* **2021**, *139*, 8821–8831.
12. Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E.; Ghasemipour, S.K.S.; Ayan, B.K.; Mahdavi, S.S.; Lopes, R.G.; et al. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *arXiv* **2022**, arXiv:2205.11487.
13. BOBLOW, D. Natural language input for a computer problem-solving system. *Sem. Inform. Proc.* **1968**, 146–226.
14. Charniak, E. Computer solution of calculus word problems. In *Proceedings of the 1st International Joint Conference on Artificial Intelligence*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1969; pp. 303–316.
15. Clark, P.; Etzioni, O. My computer is an honor student—but how intelligent is it? Standardized tests as a measure of AI. *AI Mag.* **2016**, *37*, 5–12.
16. Zhang, D.; Wang, L.; Zhang, L.; Dai, B.T.; Shen, H.T. The gap of semantic parsing: A survey on automatic math word problem solvers. *IEEE Trans. Patt. Anal. Mach. Intell.* **2019**, *42*, 2287–2305.
17. Ki, K.S.; Lee, D.G.; Gweon, G. KoTAB: Korean template-based arithmetic solver with BERT. In Proceedings of the 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), Busan, Korea, 19–22 February 2020; pp. 279–282.
18. Fletcher, C.R. Understanding and solving arithmetic word problems: A computer simulation. *Behav. Res. Methods Instrume. Comput.* **1985**, *17*, 565–571.
19. Bakman, Y. Robust understanding of word problems with extraneous information. *arXiv* **2007**, arXiv:math/0701393 [math.GM].
20. Roy, S.; Vieira, T.; Roth, D. Reasoning about quantities in natural language. *Trans. Assoc. Comput. Linguist.* **2015**, *3*, 1–13.
21. Mitra, A.; Baral, C. Learning to use formulas to solve simple arithmetic problems. In Proceedings of the Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 2144–2153.
22. Wang, Y.; Liu, X.; Shi, S. Deep neural solver for math word problems. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 845–854.
23. Wang, L.; Wang, Y.; Cai, D.; Zhang, D.; Liu, X. Translating a Math Word Problem to a Expression Tree. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 1064–1069.
24. Chiang, T.R.; Chen, Y.N. Semantically-Aligned Equation Generation for Solving and Reasoning Math Word Problems. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, 2–7 June 2019; pp. 2656–2668.
25. Griffith, K.; Kalita, J. Solving arithmetic word problems automatically using transformer and unambiguous representations. In Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 5–7 December 2019; pp. 526–532.
26. Kim, B.; Ki, K.S.; Lee, D.; Gweon, G. Point to the expression: Solving algebraic word problems using the expression-pointer transformer model. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 3768–3779.
27. Roy, S.; Roth, D. Solving General Arithmetic Word Problems. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1743–1752.
28. Koncel-Kedziorski, R.; Roy, S.; Amini, A.; Kushman, N.; Hajishirzi, H. MAWPS: A math word problem repository. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1152–1157.
29. Hosseini, M.J.; Hajishirzi, H.; Etzioni, O.; Kushman, N. Learning to solve arithmetic word problems with verb categorization. In Proceedings of the EMNLP, Doha, Qatar, 25–29 October 2014; pp. 523–533.
30. Koncel-Kedziorski, R.; Hajishirzi, H.; Sabharwal, A.; Etzioni, O.; Ang, S.D. Parsing algebraic word problems into equations. *Trans. Assoc. Comput. Linguist.* **2015**, *3*, 585–597.

31. Kushman, N.; Artzi, Y.; Zettlemoyer, L.; Barzilay, R. Learning to automatically solve algebra word problems. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Baltimore, Maryland, 22–27 June 2014; pp. 271–281.
32. Miao, S.Y.; Liang, C.C.; Su, K.Y. A Diverse Corpus for Evaluating and Developing English Math Word Problem Solvers. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 975–984.
33. Qin, J.; Lin, L.; Liang, X.; Zhang, R.; Lin, L. Semantically-Aligned Universal Tree-Structured Solver for Math Word Problems. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 3780–3789.
34. Lin, X.; Shimotsuji, S.; Minoh, M.; Sakai, T. Efficient diagram understanding with characteristic pattern detection. *Comput. Vis. Graph. Image Proc.* **1985**, *30*, 84–106.
35. Seo, M.; Hajishirzi, H.; Farhadi, A.; Etzioni, O.; Malcolm, C. Solving geometry problems: Combining text and diagram interpretation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1466–1476.
36. Alvin, C.; Gulwani, S.; Majumdar, R.; Mukhopadhyay, S. Synthesis of problems for shaded area geometry reasoning. In *Proceedings of the International Conference on Artificial Intelligence in Education*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 455–458.
37. Graepel, T.; Obermayer, K. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*; MIT Press: Cambridge, MA, USA, 2000; pp. 115–132.
38. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inform. Proc. Syst.* **2013**, *26*, 3111–3119.
39. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
40. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146.
41. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002; pp. 311–318.