Enhancing Math Word Problem Solving Using Multi-Head-Attention Mechanism

Sandip Sarkar*, Dipankar Das†, David Pinto‡
*Hijli College, India
†Jadavpur University, India
‡Computer Science of the Benem erita Universidad Aut onoma de Puebla

Abstract

Solving arithmetic word problems requires models that can effectively understand and generate mathematical equations based on textual information. However, bridging the gap between machine- understandable logic and human-readable language remains a significant challenge. While some researchers have made promising strides with limited datasets, there is a need for more comprehensive approaches. In this paper, we present two novel models for solving mathematical word problems (MWPs) involving multiple unknown variables: a multi-head attention mechanism-based model and a GRU-LSTM-based Seq2seq model. Our models are evaluated using state-of-the-art metrics, such as BLEU and ROUGE, considering factors such as the number of unknown variables and sentence structure. Through extensive experimentation on diverse datasets, we achieved impressive results and surpassed previous benchmarks. This research contributes to advancing math word problem-solving by harnessing the power of deep learning techniques and attention mechanisms. Our findings also highlight areas for future improvements and expansion in this domain.

Key Words: Math word problem, Deep learning, Seq2Seq learning, Text simplification

1 Introduction

Different factors contribute to the challenges students face when dealing with mathematical word problems. These obstacles encompass factors such as math-related anxiety, limitations in memory capacity, inadequate counting skills, language barriers, and a deficiency in problem-solving strategies. [1, 2]. Math anxiety manifests itself when dealing with numerical problems [3, 4]. Students experiencing anxiety necessitate additional time to solve math word problems, while those lacking counting and number sense skills also require extended durations for solving such problems [5, 6].

On the contrary, English assumes the status of the predominant language of instruction in a vast number of nations. Consequently, students hailing from non-English-speaking countries encounter certain challenges while grasping their academic subjects [7]. Specifically, if a student needs additional time to comprehend a math problem presented in English, a corresponding increase in time will also be essential for solving said problem. It is worth noting that math word problems often incorporate intricate vocabulary, posing difficulties in comprehension.

Creating an automated system to solve mathematical word problems poses a challenging endeavor within the realm of Natural Language Processing [8, 9]. Since the 1960s, researchers have put forward multiple methodologies aimed at creating an automatic mathematical problem-solving system [10, 11, 12, 13, 14]. The primary objective of math problem solving is to develop algorithms capable of solving math problems expressed in natural language. An automatic math solver aids students in comprehending mathematical word problems and necessitates the creation of a tutorial system. Several researchers argue that utilizing small datasets with limited variation can yield encouraging outcomes [15, 16, 17].

Mathematical word problems can be categorized into two distinct types: i) arithmetic word problems and ii) algebraic word problems [18]. Arithmetic word problems predominantly involve fundamental mathematical operations, while algebraic word problems encompass more intricate operators like square roots, exponentials, and logarithms, often involving multiple unknown variables. For the purposes of this paper, our focus will solely be on arithmetic word problems.

Table 1 shows examples of the math word problem. A word problem is a topic in mathematics that elementary school students typically resolve. Questions are presented in plain text using the symbols W_1, W_2, etc. The quantities are written as q_1, q_2, q_3 with X standing in for the unknown variable. The output of a math word problem only presents the four types of basic operators O = +, –, ×, ÷.

Our primary objective is to assess the impact of varying the unknown variable on the generation of equations. Certain researchers focus on analyzing a single unknown variable, while others examine scenarios involving multiple unknown variables. Nevertheless, there is currently a lack of comprehensive research that compares and evaluates the system's performance for both types of unknown variables. Table 4 shows the data variation of the math word problems.

This paper uses two types of deep learning approaches 1) GRU-LSTM based Seq2seq Model, 2) Multi-Head Attention Model. In addition, we evaluated our models using BLEU and ROUGE matric scores. Math word problems not only depend on the number of unknown variables but also on the structure of the sentence (i.e., simple or complex forms). In

* Corresponding author - Sandipsarkar.ju@gmail.com

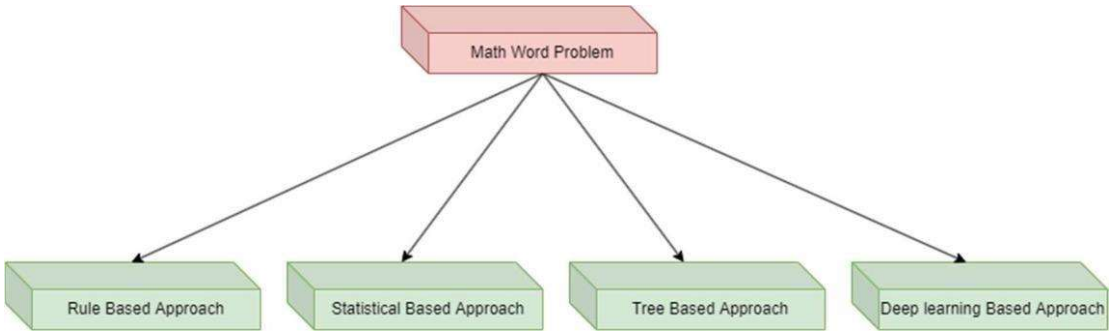

Figure 1: Categories of Math Word Problem

Problem 1	**Statement:**	Two numbers have a sum of 45 and a difference of 29. Can you find the values of these two numbers?
	Equation:	x+y=45.0 x-y=29.0
	Solution:	37, 8
Problem 2	**Statement:**	The student-teacher ratio for Washington High was reported to be 12 to 1. If there are 15 teachers, then how many students are there?
	Equation:	12*15=x*1
	Solution:	180.0

Table 1: Problem Definition

observation, we also give an overview of the reasons behind the poor performance of our system on some of the datasets.

The structure of our paper is outlined in the following manner. In Section 2, we conduct a thorough literature survey to examine existing research in the field. This provides a solid foundation for our study. In Section 3, we provide a detailed description of the dataset used, outlining its composition and characteristics. Moving forward, Section 4 presents our approach, discussing the methodologies and techniques employed to address the problem. Section 5 focuses on the training process, providing insights into the specific procedures and considerations taken. In Section 6, we present the experimental results of our approach. Additionally, Section 7 delves into our observations, discussing any limitations or drawbacks identified in our system. Finally, in Section 8, we conclude our work, summarizing the key findings and contributions made throughout the study, thereby bringing our research to a conclusive end.

2 Literature Survey

The advancement of systems designed to address arithmetic word problems has garnered significant attention in recent years. These methods can be categorized into four main groups: (i) Rule-based Approach, (ii) Statistical Approach, (iii) Tree-based Approach, and (iv) Neural Network-Based Approach. Each of these approaches presents distinct methodologies and techniques, contributing to the diverse landscape of arithmetic word problem-solving methodologies.

2.1 Rule based Approach

There are certain limitations associated with rule-based approaches to solve math word problems. One major drawback lies in its restricted coverage and the complexity associated with its implementation, as indicated by previous research [19, 20]. The rule-based approach may struggle to handle a wide range of math word problems effectively, and its intricate implementation can pose challenges in terms of development and maintenance. Moreover, scalability becomes a concern when dealing with large datasets. Notably, the WORDPRO system, proposed in 1985, introduced a rule-based approach for solving math word problems, incorporating four distinct schemes: change-in, change-out, combine, and compare [21]. Another noteworthy system, called ROBUST, was developed by Bakman specifically to address free-format multi-step arithmetic word problems [22].

2.2 Statistical Approach

In order to solve math word problems, statistical learning-based methods have been proposed since 2014 [23, 24]. By leveraging statistical techniques, such as machine learning algorithms, pattern recognition, and data analysis, this approach aims to extract relevant information, identify problem structures, and make informed predictions or decisions to solve math word problems efficiently. It offers a promising avenue for enhancing problem-solving capabilities and addressing the complexities often encountered in mathematical word problems. By Hosseini and others the open-domain nature of algebraic

Dataset	Problem Statement	Equations	Solutions
number word std	Find two numbers whose sum is 33 and one of the numbers is two times as large as the other?	x + y = 33, x = 2*y	x=22 , y =11
DRAW-1K	kevin 's sister mary is 5 years younger than he is . the total of their ages is 33 years . how old is kevin ?	x+y=33, y=x-5	x=19, y=14
Math23K	**Original Text:** 一种电视，电视现在售价450 元，比原来便宜10% ，比原来便宜多少元? **Translated Text:** A kind of TV. The price of TV is 450 yuan now, which is 10% cheaper than before. How much is it cheaper than before?	x=450/(1-10%)-450	x = 50

Table 2: Description of Dataset

word problems is addressed by learning verb categorization from training data [25]. The operator classification method is used by the author of [24] to resolve math word problems. The operator classifier divides the math word problem by operator

$op \in (+, -, *, /)$.

2.3 Tree based Approach

The utilization of tree structures is central to the tree-based approach for analyzing and solving mathematical problems presented in text format. This approach represents the problem as a tree, with nodes representing various problem components (such as numbers, operators, and variables), and edges representing the connections between them. By employing this method, a systematic understanding of the problem and the generation of solutions are made possible. Binary tree structures can be used to represent arithmetic expressions [26, 27]. In a binary tree structure, operators with higher priority are positioned at lower levels, and the operators with the lowest priority are found at the tree's root[28, 29].

2.4 Neural Network Based Approach

Since 2012, deep learning has been applied to a variety of natural language processing tasks, such as Question Answering [30], text simplification [31], sentiment analysis [32], machine translation [33] as well as math word problem [34, 35, 36, 37]. A system that can produce equation templates from a mathematical statement was developed by Wang in 2017 [38]. Next, a model that adds an attention-based mechanism to the prior model was presented by Huang [39]. Deep learning's main benefit is retrieval its ability to select features efficiently without human assistance [40]. On the other hand, it has one drawback: its learning process necessitates a lot of data [41].

3 Dataset

Math word problem datasets commonly suffer from limitations in terms of size and the variety of mathematical scenarios they cover. In our research, we sought to overcome this by meticulously selecting three datasets: Dolphin T2 Final, DRAW-1, and Math23K. Among these, Math23K and Dolphin T2 Final exhibited the most promising characteristics for math word problems. In the subsequent sections, we will provide comprehensive insights into these datasets, emphasizing their composition and significance in our research. The example of each dataset is given in Table 2. Similarly, Table 3 shows the statistics of each dataset.

3.1 Dolphin T2 Final

Dolphin T2 Final is the part of Dolphin18K dataset. [1] has a greater variety of problem types. Their main objective is to use simple mathematics to build a large dataset [42]. The mathematics category of the Yahoo! Answers website provided the dataset. The raw problem text and one or more solutions make up a math problem.

The Dolphin T2 Final dataset focuses on math word problems and is composed of 831 problem instances that have been submitted by users on the popular community driven platform Yahoo! Answers [43].

3.2 DRAW-1K

The Diverse Algebra Word Problem Set (DRAW) dataset [2] encompasses a collection of 1000 algebra word problems. Notably, DRAW surpasses the Alg-514 dataset with its tenfold increase in equation templates and double the number of problem instances. This significant expansion ensures that DRAW exhibits a more realistic representation of real-world problem scenarios. Moreover, DRAW holds the distinction of being the first dataset to provide alignments between equation coefficients and corresponding numbers within the problem text, along with annotated templates. These unique features make

[1]https://msropendata.com/datasets/f0e63bb3-717a-4a53-aa79-da339b0d7992

[2]https://www.microsoft.com/en-us/download/details.aspx?id=52628

Dataset	Problems	Sentences	Percentage of (> 1)Unknown Variables	Percentage of (1) Unknown Variable	Question Length	Equation Length
Dolphin T2 Final	831	831	26.23%	73.76%	75.64	12.93
DRAW-1K	1000	2330	74.5%	25.5%	103.90	12.27
Math23K	23162	70125	0%	100%	70.76	13.944

Table 3: Statistics of Dataset

Dataset Variation	Problem Statement	Equations
Single Variable	The sum of two consecutive whole numbers is 27. Find the numbers.	x+(x+1)=27
	Currently, a person is 35 years old, and his son is 6 years old. After how many years will the son be half the age of her father?	0.5*(35+x)=(6+x)
	To create a solution with a 35% acid concentration, how many liters of a 60% acid solution need to be combined with 11 liters of a 15% acid solution?	0.01*60*x+0.01*15*11=0.01*35*(x+11)
Multiple Variable	A wire with a length of 160 cm needs to be divided into two sections, with one part being 25 cm longer than the other. What is the length of each part?	x+y=160, x=y+25
	Two trains commence their journeys from towns that are 192 miles apart, moving towards each other on parallel tracks. After 1.6 hours, they cross paths. If one train travels at a speed 10 mph faster than the other, determine the speed of each train	x-y=10, (x+y)*1.6=192
	Rajesh is conducting an experiment in the laboratory where he is combining two alcohol-containing solutions. He uses a quantity of Solution A that is 500 milliliters less than Solution B. Solution A has an alcohol concentration of 16%, while Solution B has a concentration of 10%. If the resulting mixture contains 76 milliliters of pure alcohol, how many milliliters of Solution A does Rajesh use?	0.01*10*x+0.01*16*y=76, (x-500)=y

Table 4: Data Variation in Math Word Problem

DRAW an invaluable resource for advancing research in the field [44, 45].

3.3 Math23K

The Math23K dataset is predominantly derived from various online educational websites, making it a comprehensive resource for math word problems [38, 46]. Given the requirement of a large dataset for training deep learning models, Math23K's impressive collection of 23,161 problem instances makes it an ideal choice. This dataset is particularly well-suited for elementary school students, as it aligns with their educational level and curriculum. The compilation of Math23K involved the use of a rule-based extraction mechanism, enabling the retrieval of both problem statements and their corresponding solutions. Notably, the math word problems in Math23K can be effectively solved using linear algebra expressions that involve a single unknown variable.

4 System Architecture

The dataset has been thoroughly described in the preceding section. Our proposed models performed on three different datasets. All datasets, with the exception of Math23k, contain a number of unknown variables, as was already mentioned while the Dolphin T2 Final and DRAW 1K datasets contains multiple unknown variables. Table 3 shows the percentage of math word problems with one unknown variable and math word problems with multiple unknown variables in the dataset. On the other hand, Math23K only contains math word problems that contain one unknown variable.

To deal with we present two innovative models that offer

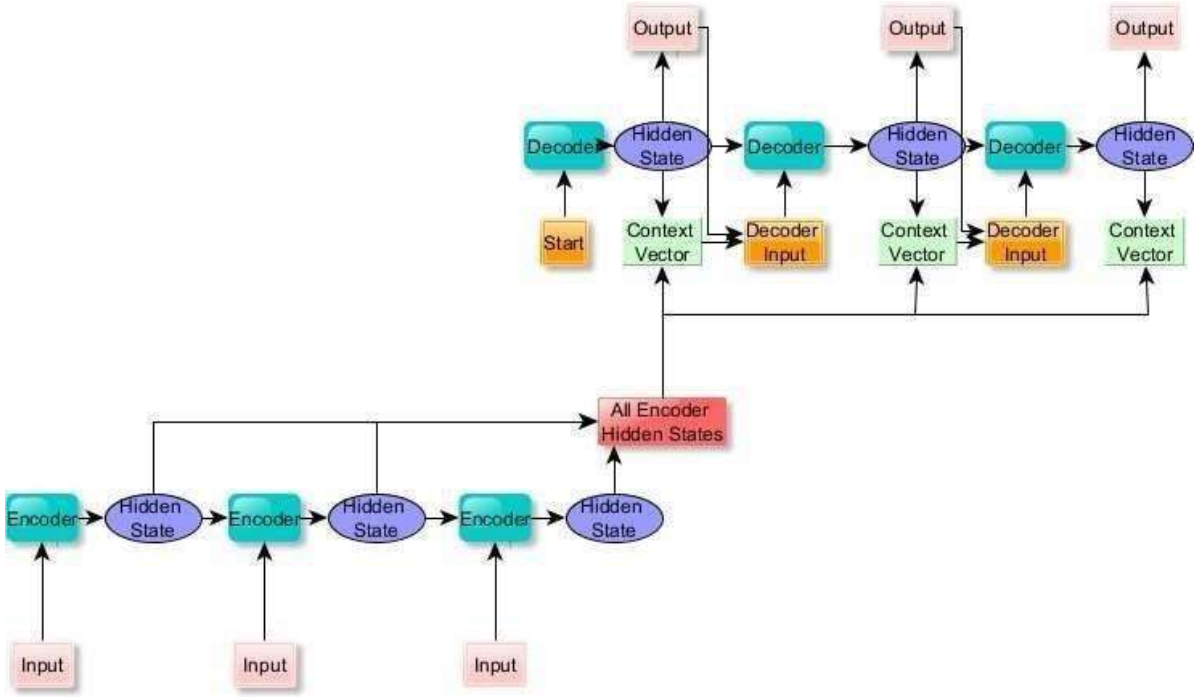

Figure 2: GRU-LSTM based Seq2seq Model

novel solutions to the challenges posed by mathematical word problems (MWPs) featuring multiple unknown variables. Our research introduces a cutting-edge approach utilizing a multi-head attention mechanism -based model, as well as a state-of-the-art Seq2seq model that combines GRU and LSTM. These models represent significant advancements in the field and provide promising avenues for effectively addressing complex MWPs. Table 4 illustrates various types of math word problems.

4.1 GRU-LSTM based Seq2seq Model

Our baseline model predominantly revolves around sequence -to-sequence modeling, which serves as its fundamental framework. Notably, it employs bidirectional GRU cells for the encoding phase and LSTM cells for the decoding phase. To preprocess the equation string, we employ a specific procedure. Initially, each character within the string is separated by a space. Subsequently, all characters in the math statement are converted to lowercase, while punctuation marks are accompanied by spaces inserted between them. As the final step, any remaining extra spaces and digits are removed from the processed equation. For the encoder component, we utilize a Bi-LSTM (Bidirectional Long Short-Term Memory) model, featuring 256 hidden units, 32 embedding dimensions, and a batch size of 32. Additionally, the LSTM cell within the model is configured with the same set of parameters.

Furthermore, the training process incorporates the Adam optimizer with a learning rate of 0.001, which aids in optimizing the model's performance. The training specifications include 10 epochs, a batch size of 100, and a dropout rate of 0.5. These settings ensure effective model training and help mitigate overfitting, leading to improved generalization capabilities. The architecture of the GRU-LSTM based Seq2seq Model is depicted in Figure 2.

4.2 Multi-Head Attention Mechanism-based Model

The Multi-head Attention module is an advanced component that operates in a complex manner through attention mechanisms. It iterates simultaneously and repeatedly within the attention mechanism. This process carefully combines the independent attention outputs using linear combinations, resulting in a meticulously crafted dimension. By using multiple attention heads, a system can focus on different elements in a sequence, making it easier to prioritize longer-term or shorter-term dependencies as needed. The Transformer architecture follows a hierarchical structure, using stacked self-attention and fully connected layers in both the encoder and decoder. This design allows for thorough modeling of relationships and information flow at various levels, resulting in improved understanding and generation of sequences. Figure 3 visually shows the architectural design, with the left and right portions representing different aspects. This graphical representation helps us understand how the components are arranged and interact with each other within the system.

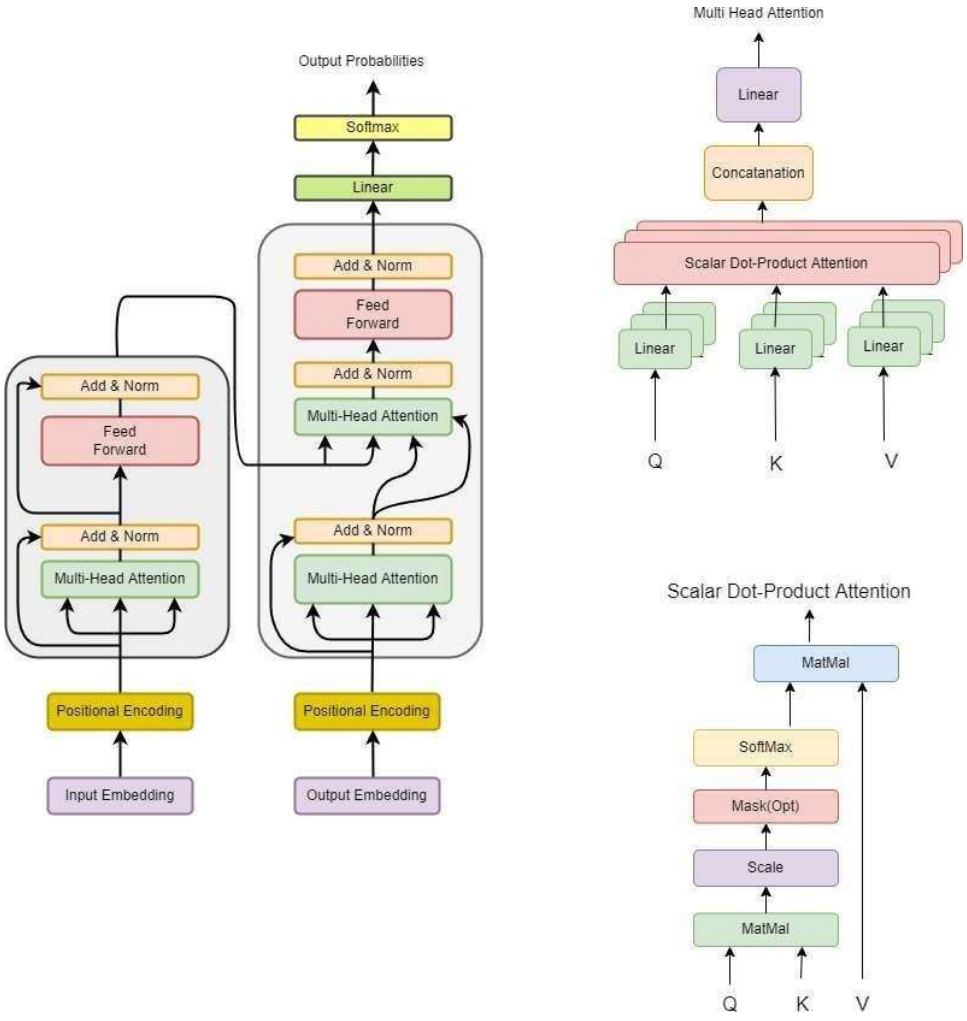

Figure 3: Multi-Head Attention Model

4.2.1 Encoder and Decoder Stacks

The Encoder and Decoder stacks have significant importance in the Multi-head Attention mechanism, which serves as a fundamental building block in the Transformer architecture widely utilized across various natural language processing applications. The Encoder and decoder of our multi-head attention mechanism-based model consist of 4 identical layers. Each layer of encoder and decoder is divided into two sub-layers. There are two components involved. The initial component is a self-attention mechanism with multiple heads, while the subsequent component is a positioned fully connected feed-forward network. Apart from the two sub-layers already present, the decoder introduces an additional sub-layer to each layer of the encoder.

The sub-layer that has been newly introduced conducts multi-head attention operations on the output of the encoder stack, thereby enhancing the model's ability to capture intricate relationships and dependencies within the encoded information. Within both the encoder and decoder components, once a residual connection has been applied around each of the two sub-layers, the subsequent step involves the application of layer normalization. To provide a concise explanation, we can express each sub-output layer as LayerNorm(a + Sublayer(a)), where the term Sublayer(a) denotes the function performed by the sub-layer itself.

Moreover, within the decoder stack, we introduce modifications to the self-attention sub-layer in order to enforce a constraint that prevents positions from attending to future positions.

4.2.2 Scaled Dot-Product Attention

Scaled dot-product attention consists of a number of key-value pairs that map a query to an output. To determine the weights on the values, we compute the dot products of the query with each key and divide each result by d_k. The obtained weights are normalized using the Softmax function.

For the Query, Key, and Value, there are three different linear layers. The weights of each linear layer are unique. These linear layers process the input to create the Q, K, and V matrices. The equation is presented as follows:

$$\text{Attention(Q, K, V)} = softmax \frac{QK^T}{\sqrt{d_k}} V \quad (1)$$

4.2.3 Multi-Head Attention

The Attention module of the Transformer carries out its computations in a repetitive and parallel manner, which are known as Attention Heads. The Attention module divides the Query, Key, and Value parameters into N parts and processes each split separately through individual Heads. After conducting all the relevant Attention calculations, a final Attention score is generated by aggregating the results of these computations. By employing a method called "Multi-head attention," the Transformer enhances its ability to capture diverse relationships and intricate details associated with each word during the encoding process.

$$head_i = attention(QW_i^Q, KW_i^K, VW_i^Q) \quad (2)$$

$$Multihead(Q, K, V) = concat(head_1, \ldots, head_m)W^o \quad (3)$$

Where the projections are parameter matrices $W_i^Q R^{d_{model} \times d_k}$, $W_i^K R^{d_{model} \times d_k}$, $W^V{}_i R^{d_{model} \times d_v}$ and $W_O R^{hd_v \times dmodel}$.

4.2.4 Self Attention

The input sequence is broken down into three parts during self-attention: queries, keys, and values. These components undergo linear transformations to generate representations. Attention scores are then computed to determine the significance of each word or token to others, based on the relationship between queries and keys. By dividing the values by the attention scores, a weighted sum is obtained, representing the final representation of each word or token.

The utilization of multiple heads in self-attention enables the model to capture a variety of relationships among words or tokens. Each head focuses on a distinct area of the input, allowing the model to process long-range dependencies and gather both local and global data with efficiency.

5 Results

Deep learning has found its way into various facets of our everyday lives, with certain applications demanding significant computational power and extensive datasets. Parallel computing has emerged as a solution to handle such demanding scenarios. One prominent example of parallel processing is the utilization of graphics processing units (GPUs). In our specific task of generating mathematical equations, we employed GPUs to leverage their immense processing capabilities.

It is already mentioned that Multi-head mechanism-based model has four Encoder-Decoder layers and a 128-embedding dimension for input/output. We have four self-attention heads with a batch size of 64. For the Math23k, number word std, DRAW-1K, and Dolphin 18K datasets, models are trained in 30, 50, 65, and 55 epochs respectively. We employ the Adam optimizer to fine-tune the models, utilizing a learning rate of 0.001.

Cloud-based solutions are popular because it do not require system maintenance or configuration. There are many companies that provide cloud-based services such as Amazon, Google, Azure, and Intel. We employed Google Colaboratory (also referred to as Colab) for our experiment. Colab is a cloud-centric platform built upon Jupyter Notebooks [47]. Jupyter is a tool that is open-source and accessible through a web browser. It is very easy to use and can be used both locally and in the cloud. Google Colaboratory has some limitations. After 12 hours we need to configure the Colab.

Our proposed Multi-Head attention mechanism produced promising results when compared to other rule-based and statistical methods. Our model, on the other hand, performs well on the large complex dataset. It can generate a wide range of equations. Both models perform well on the Math23K dataset but poorly when the math word problem contains two unknown variables.

As mentioned earlier, in order to address math word problems featuring multiple unknown variables, we partitioned the dataset (consisting of Dolphin T2 Final and DRAW 1K) into two subsets: one containing math problems with a single unknown variable, and the other containing math problems with multiple unknown variables. The resulting output generated by our system can be observed in Table 6, providing a comprehensive overview of the outcomes obtained through our approach.

On the other hand, stop words are generally considered to have minimal significance in sentences. Consequently, our model exhibits better performance when we employ customized stop words. The generated output produced by our system is presented in Table 6, showcasing the results achieved through our approach.

For evaluation, we have used two evaluation matrices. A detailed description of those evaluation matrices is given below.

BLEU: In comparison to using human judgments, automatic machine translation evaluation metrics offer a quicker and less expensive method of assessing translation quality. The benchmark evaluation metric (BLEU) for machine translation (MT) has been shown to have a fair amount of agreement with human judges despite being straightforward and independent of language[48]. It is frequently employed as a loss function for discriminative training in addition to being used in the evaluation. BLEU was created to evaluate MT output across a wide range of references and sizeable documents.

ROUGE: It stands for Recall-Oriented Understudy for Gisting Evaluation, is commonly employed as a reference to the evaluation metric used for assessing the quality of automatic summaries. It has several automated evaluation techniques that gauge how similar the summaries are to one another [49]. The measurements count the number of units that are provided by the computer-generated equations to be evaluated and the ideal equations, such as n-grams, word sequences, and word pairs.

In our experimental setup, we divided the dataset into two distinct subsets, allocating 80% of the data for training purposes and reserving the remaining 20% for testing. For evaluation, we used the BELU score. Table 5 provides a detailed summary of the outcomes for those datasets. On the Math23K dataset, our model performs exceptionally well.

6 Observation

Our proposed Multi-Head attention mechanism produced promising results when compared to other rule-based and statistical methods. In contrast, our model demonstrates strong performance when dealing with extensive and intricate datasets. It can generate a wide range of equations. Both models perform better on the Math23K dataset but poorly when the math word problem contains two unknown variables.

We analyze the observations based on the dataset, considering that our models yield distinct outputs for each dataset. This discussion allows us to gain insights into the performance and variations exhibited by our models when confronted with different datasets.

Furthermore, it is widely acknowledged that stop words have limited significance in sentences. Consequently, our model demonstrates enhanced performance when customized stop words are employed. The output generated by our system is presented in Table 6, showcasing the results achieved through this utilization. The subsequent sections discuss various observations based on the dataset, providing a comprehensive analysis of the different insights obtained from our study.

6.1 DRAW-1K

The DRAW-1K dataset contains 1,000 problems and 2,330 sentences. A significant majority, about 74.5%, of the problems in this dataset involve more than one unknown variable, while only 25.5% have a single unknown variable. The average question length is relatively longer at 103.90, suggesting that the questions in this dataset are more detailed or complex. Similar to the Dolphin T2 Final dataset, the equation length is relatively short, averaging at 12.27 characters.

Model 3 demonstrates superior performance in the DRAW-1K dataset when handling multiple variables compared to single variables alone. Its ability to effectively incorporate and analyze multiple variables leads to more accurate and robust outcomes. Furthermore, when considering the entire dataset, Model 3 achieves its best results by leveraging both single and multiple variables. By considering the full spectrum of available information, the model gains a comprehensive understanding of the data, resulting in optimal performance.

6.2 Dolphin T2 Final

Dolphin T2 Final dataset consists of 831 problems and 831 associated sentences. Approximately 26.23% of the problems in this dataset involve more than one unknown variable, while 73.76% have only one unknown variable. The average question length is 75.64, indicating that the questions in this dataset tend to be relatively concise. The equation length is relatively shorter with an average of 12.93 characters.

Table 5 shows the effectiveness of Model 3 in addressing both single-variable and multiple-variable problems within the Dolphin T2 Final dataset. Model 2 also showcases the notable performance, while Model 1 needs some improvement. Table 3 also shows that Model 3 performs better for the single variable variation in the Dolphin T2 Final dataset compared to the multiple variables and combined datasets. This observation suggests that Model 3 is more effective in generating accurate and relevant responses when there is only a single variable involved in the input data.

Dataset Name	Dataset Variation	Model	BLEU	ROUGE-1			ROUGE-2		
				r	p	f	r	p	f
DRAW 1K	Single Variable	**Model 1**	0.07	0.3002	0.3021	0.3142	0.1253	0.1224	0.1178
		Model 2	0.17	0.5012	0.5212	0.5112	0.2122	0.2431	0.2152
		Model 3	**0.22**	**0.550**	**0.5535**	**0.5346**	**0.2485**	**0.2890**	**0.2588**
	Multiple Variable	**Model 1**	0.21	0.4236	0.2113	0.2242	0.2315	0.2414	0.2486
		Model 2	0.38	0.5524	0.5621	0.5412	0.4215	0.4125	0.4211
		Model 3	**0.40**	**0.5700**	**0.5881**	**0.5760**	**0.4558**	**0.4358**	**0.4358**
	Combined Dataset	**Model 1**	0.20	0.4256	0.4235	0.4351	0.2453	0.2581	0.5413
		Model 2	0.34	0.5723	0.5653	0.5692	0.4122	0.4089	0.4156
		Model 3	**0.41**	**0.6198**	**0.6114**	**0.6029**	**0.4227**	**0.4187**	**0.4187**
Dolphin T2 Final	Single Variable	**Model 1**	0.35	0.5927	0.5837	0.5892	0.4567	0.4628	0.4582
		Model 2	0.68	0.7512	0.7215	0.7304	0.6033	0.5728	0.5641
		Model 3	**0.71**	**0.7738**	**0.7434**	**0.7562**	**0.6231**	**0.5734**	**0.5924**
	Multiple Variable	**Model 1**	0.31	0.5584	0.5428	0.5872	0.4235	0.4158	0.4251
		Model 2	0.47	0.7311	0.7135	0.7012	0.5145	0.5042	0.5095
		Model 3	**0.51**	**0.7823**	**0.7224**	**0.7499**	**0.5430**	**0.5107**	**0.5244**
	Combined Dataset	**Model 1**	0.52	0.7952	0.7825	0.7836	0.5825	0.5721	0.5821
		Model 2	0.64	0.7152	0.7145	0.7288	0.5672	0.5245	0.5289
		Model 3	**0.67**	**0.7577**	**0.7327**	**0.7425**	**0.6022**	**0.5701**	**0.5811**
Math 23K	Single Variable	**Model 1**	0.67	0.7125	0.7254	0.7173	0.5897	0.5862	0.5729
		Model 2	0.91	0.8521	0.8511	0.8580	0.6127	0.6158	0.6156
		Model 3	**0.92**	**0.8769**	**0.8712**	**0.8721**	**0.5990**	**0.5920**	**0.5939**

Table 5: Result of Our Models (i.e. Model 1: GRU-LSTM based Seq2seq Model, Model 2: Multi-Head Attention Mechanism Based Model, Model 3: Multi-Head Attention Mechanism Based Model without stop word+ Comparison with other model)

6.3 Math23k

The Math23K dataset is the largest among the three, consisting of 23,162 problems and a substantial 70,125 sentences. All problems in this dataset involve only a single unknown variable. The average question length is 70.76, indicating moderately sized problem statements. The equation length is slightly longer compared to the other datasets, averaging 13.944 characters.

Model 3 consistently gives better results than the other models in terms of BLEU, ROUGE-1, and ROUGE-2 scores, indicating its superior performance in generating accurate solutions for Math23K. Model 2 also performs exceptionally well, achieving high scores across all evaluation metrics.

The complexity of sentence structures within math word problems poses a challenge for our system's recognition capabilities, leading to less satisfactory results on that specific dataset.

On the other hand, the following complex sentence is also presented in the math word problem.

Solving math word problems becomes challenging when they involve complex forms. In Table 7, we provide additional examples of math word problems that exhibit complex structures. Our system's performance is influenced by the length of the problem statement. Furthermore, when confronted with extensive equations, our system may struggle to generate the correct equation.

Addressing these challenges is an important area for future work. Developing techniques to handle math word problems with complex forms will be crucial in improving the system's performance. Additionally, devising strategies to handle longer problem statements and large equations will contribute to more accurate equation generation.

By focusing on these areas of improvement, we can enhance the system's ability to tackle complex math word problems, irrespective of their length or equation size.

7 Conclusion

We proposed a seq2seq model that uses a Multi-Head attention mechanism to generate equations from math word problems. After conducting experiments on four renowned math word problem datasets, the obtained results provide compelling evidence that the proposed model surpasses the performance of the state-of-the-art statistical model. Notably, our model exhibits a greater level of complexity within the realm of math word problems, primarily due to its ability to effectively handle a substantial quantity of unknown variables. We can still improve the system's accuracy in some areas. Moving forward, our aspirations lie in expanding upon this research endeavor to delve into the realm of generating nonlinear equations. Furthermore, we intend to apply these techniques in a variety of domains related to word problems, such as physics, chemistry, and others.

DRAW 1K	**Single Variable**	**Problem**	A number added to 6 is equal to 30 less than four times the number. what is the number.
		Actual Output	6 + x = 4 * x - 30
		Predicted Output	8 + x = 6 x + 24
	Multiple Variable	**Problem**	the larger of two numbers is 7 more than three times the smaller number. When the smaller number is subtracted from the larger number, the result is 13. find the two numbers.
		Actual Output	x = 3 * y + 7 , x - y = 13
		Predicted Output	x = 5 * y + 7 , x - y = 45
	Combined Dataset	**Problem**	Jim can fill a pool carrying buckets of water in 30 minutes. Sue can do the same job in 45 minutes. Tony can do the same job in 90 minutes. how quickly can all three fill the pool together?
		Actual Output	x * (1 / 30 + 1 / 45 + 1 / 90) = 1
		Predicted Output	x * (1 / 25 - 1 / 52 + 1 / 2) = 1
Dolphin T2 Final	**Single Variable**	**Problem**	The sum of two consecutive integers is 237 . Find the two integers.
		Actual Output	x + (x + 1.0) = 237.0
		Predicted Output	x + (x + 1.0) = 125.0
	Multiple Variable	**Problem**	The sum of two numbers is 52. The larger number is three times the smaller number . Find the larger number
		Actual Output	x + y = 52.0 , y = 3.0 * x
		Predicted Output	x + y = 45.0 , y = 3.0 * x
	Combined Dataset	**Problem**	there are 3 consecutive integers. the sum of the first two integers is 16 more than the third. Find integers
		Actual Output	(x - 1.0) + x = (x + 1.0) + 16.0
		Predicted Output	(x - 1.0) + x = (x + 1.0) + 14.0
Math 23k	**Single Variable**	**Problem**	A certain kind of book is priced at 72 yuan per set of 6 . The bookstore sold 18 sets of such books yesterday. How much did they sell for?
		Actual Output	x = 72 * 18
		Predicted Output	x = 72 * 18

Table 6: Output Produced by Multi-Head Attention Mechanism

This paper addresses the challenges posed by math word problems containing more than two unknown variables. These complex problems can be particularly demanding to solve effectively. As is customary, math word problems often involve the four fundamental operations (+, -, /, *). To gain further insights, we plan to partition the dataset based on these fundamental operations. Subsequently, we conduct an in-depth examination of the performance of each partitioned dataset.

On the other hand, we also looked at how the system would perform in the attention model if we split the joint sentence and divided the multiple sentences into separate sentences.

Conflict of interest statement

The authors state that they do not have any conflicts of interest to disclose. All co-authors have reviewed and approved the manuscript's content, and no financial interests are to be reported. The authors affirm that the submission represents original work and is not being considered for publication elsewhere.

References

[1] M. Hickendorff, The demands of simple and complex arithmetic word problems on language and cognitive resources, Frontiers in Psychology 12 (2021). doi:10.3389/fpsyg.2021.727761.

	Problem Statement	Actual Output	Predicted Output
Case 1	A salesperson at a machine company is offered an incentive plan. They earn a commission of $40 for each machine they sell. Additionally, the commission increases by $0.04 for every machine sold beyond $600. The salesperson wants to know how many machines they need to sell to reach a total commission of $30,800.	30800=40*x+0.04*(x-600)	30800=40*x+0.04*(x-600)
Case 2	The total of a two-digit number is 9. When the digits are swapped, the resulting number is six times the sum of the original digits. Determine the original number.	smaller+larger=9, 10 * smaller + larger = 6 *(smaller + larger)	x + y = 9, 10 * x + y = 4 x + 5y

Table 7: Challenging Math Word Problem

URL https://www.frontiersin.org/articles/10.3389/fpsyg.2021.727761

[2] Y. Hong, Q. Li, D. Ciao, S. Huang, S. Zhu, Learning by fixing: Solving math word problems with weak supervision, CoRR abs/2012.10582 (2020). arXiv:2012.10582.
URL https://arxiv.org/abs/2012.10582

[3] S. A. Crossley, S. Karumbaiah, J. Ocumpaugh, M. J. Labrum, R. S. Baker, Predicting math identity through language and click-stream patterns in a blended learning mathematics program for elementary students, Journal of Learning Analytics 7 (1) (2020) 19–37. doi:10.18608/jla.2020.71.3.
URL https://learning-analytics.info/index.php/JLA/article/view/6585

[4] G. Daroczy, M. Wolska, W. D. Meurers, H.-C. Nuerk, Word problems: a review of linguistic and numerical factors contributing to their difficulty, Frontiers in Psychology 6 (2015). doi:10.3389/fpsyg.2015.00348.
URL https://www.frontiersin.org/articles/10.3389/fpsyg.2015.00348

[5] Z. Liang, X. Zhang, Solving math word problems with teacher supervision, in: Z.-H. Zhou (Ed.), Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, International Joint Conferences on Artificial Intelligence Organization, 2021, pp. 3522–3528, main Track.

[6] A. Pathak, P. Pakray, S. Sarkar, D. Das, A. F. Gelbukh, Mathirs: Retrieval system for scientific documents, Computacio´n y Sistemas 21 (2) (2017).
URL http://www.cys.cic.ipn.mx/ojs/index.php/CyS/article/view/2743

[7] S. Sarkar, D. Das, P. Pakray, D. Pinto, Generating equations from math word problem using deep learning approach, in: S. Mukhopadhyay, S. Sarkar, P. Dutta, J. K. Mandal, S. Roy (Eds.), Computational Intelligence in Communications and Business Analytics, Springer International Publishing, Cham, 2022, pp. 252–259.

[8] A. Mukherjee, U. Garain, A review of methods for automatic understanding of natural language mathematical problems, Artif. Intell. Rev. 29 (2008) 93–122. doi:10.1007/s10462-009-9110-0.

[9] S. Powell, Solving word problems using schemas: A review of the literature, Learning disabilities research practice : a publication of the Division for Learning Disabilities, Council for Exceptional Children 26 (2011) 94–108. doi:10.1111/j.1540-5826.2011.00329.x.

[10] S. S. Sundaram, D. Khemani, Natural language processing for solving simple word problems, in: Proceedings of the 12th International Conference on Natural Language Processing, NLP Association of India, Trivandrum, India, 2015, pp. 394–402.
URL https://aclanthology.org/W15-5955

[11] B. Siyam, A. A. Saa, O. Alqaryouti, K. Shaalan, Arabic arithmetic word problems solver, Procedia Computer Science 117 (2017) 153–160, arabic Computational Linguistics. doi:https://doi.org/10.1016/j.procs.2017.10.104.
URL https://www.sciencedirect.com/science/article/pii/S1877050917321610

[12] K. Griffith, J. Kalita, Solving arithmetic word problems automatically using transformer and unambiguous representations, CoRR abs/1912.00871 (2019).

arXiv:1912.00871.
URL http://arxiv.org/abs/1912.00871

[13] K. Yokoi, A. Aizawa, An approach to similarity search for mathematical expressions using mathml, 2009.

[14] Q. Wu, Q. Zhang, X. Huang, Automatic math word problem generation with topic-expression co-attention mechanism and reinforcement learning, IEEE/ACM Transactions on Audio, Speech, and Language Processing 30 (2022) 1061–1072. doi:10.1109/TASLP.2022.3155284.

[15] B.-S. Ilany, Language and mathematics: Bridging between natural language and mathematical language in solving problems in mathematics, Creative Education 01 (2010) 138–148. doi:10.4236/ce.2010.13022.

[16] S. Mandal, S. K. Naskar, Classifying and solving arithmetic math word problems—an intelligent math solver, IEEE Transactions on Learning Technologies 14 (1) (2021) 28–41. doi:10.1109/TLT.2021.3057805.

[17] L. Verschaffel, S. Schukajlow, J. Star, W. Dooren, Word problems in mathematics education: a survey, ZDM 52 (2020) 1–16. doi:10.1007/s11858-020-01130-4.

[18] D. Zhang, L. Wang, N. Xu, B. T. Dai, H. T. Shen, The gap of semantic parsing: A survey on automatic math word problem solvers, CoRR abs/1808.07290 (2018). arXiv:1808.07290.
URL http://arxiv.org/abs/1808.07290

[19] S. Acharya, R. Basak, S. Mandal, Solving arithmetic word problems using natural language processing and rule-based classification, International Journal of Intelligent Systems and Applications in Engineering 10 (2022) 87–97. doi:10.18201/ijisae.2022.271.

[20] A. Amini, S. Gabriel, S. Lin, R. Koncel-Kedziorski, Y. Choi, H. Hajishirzi, Mathqa: Towards interpretable math word problem solving with operation-based formalisms, CoRR abs/1905.13319 (2019). arXiv:1905.13319.
URL http://arxiv.org/abs/1905.13319

[21] C. Fletcher, Understanding and solving arithmetic word problems: A computer simulation, Behavior Research Methods 17 (5) (1985) 565–571. doi:10.3758/BF03207654.

[22] Y. Bakman, Robust understanding of word problems with extraneous information (2007). doi:10.48550/ARXIV.MATH/0701393.
URL https://arxiv.org/abs/math/0701393

[23] N. Kushman, Y. Artzi, L. Zettlemoyer, R. Barzilay, Learning to automatically solve algebra word problems, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Baltimore, Maryland, 2014, pp. 271–281. doi:10.3115/v1/P14-1026.
URL https://aclanthology.org/P14-1026

[24] K. Wang, Z. Su, Dimensionally guided synthesis of mathematical word problems, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16, AAAI Press, 2016, p. 2661–2668.

[25] M. J. Hosseini, H. Hajishirzi, O. Etzioni, N. Kushman, Learning to solve arithmetic word problems with verb categorization, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 523–533. doi:10.3115/v1/D14-1058.
URL https://aclanthology.org/D14-1058

[26] T. Liu, Q. Fang, W. Ding, Z. Wu, Z. Liu, Mathematical word problem generation from commonsense knowledge graph and equations, CoRR abs/2010.06196 (2020). arXiv:2010.06196.
URL https://arxiv.org/abs/2010.06196

[27] D. Alvarez-Melis, T. S. Jaakkola, Tree-structured decoding with doubly-recurrent neural networks, in: International Conference on Learning Representations, 2017.
URL https://openreview.net/forum?id=HkYhZDqxg

[28] Q. Liu, W. Guan, S. Li, D. Kawahara, Tree-structured decoding for solving math word problems, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 2370–2379. doi:10.18653/v1/D19-1241.
URL https://aclanthology.org/D19-1241

[29] L. Wang, Y. Wang, D. Cai, D. Zhang, X. Liu, Translating a math word problem to a expression tree, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 1064–1069. doi:10.18653/v1/D18-1132.
URL https://aclanthology.org/D18-1132

[30] D. Singh, K. Suraksha, S. Nirmala, Question answering chatbot using deep learning with nlp, in: 2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 2021, pp. 1–6. doi:10.1109/CONECCT52877.2021.9622709.

[31] S. Sarkar, D. Das, P. Pakray, D. Pinto, A hybrid sequential model for text simplification, in: N. Priyadarshi,

S. Padmanaban, R. K. Ghadai, A. R. Panda, R. Patel (Eds.), Advances in Power Systems and Energy Management, Springer Singapore, Singapore, 2021, pp. 33–42.

[32] P. C. Shilpa, R. Shereen, S. Jacob, P. Vinod, Sentiment analysis using deep learning, in: 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), 2021, pp. 930–937. doi:10.1109/ICICV50876.2021.9388382.

[33] S. P. Singh, A. Kumar, H. Darbari, L. Singh, A. Rastogi, S. Jain, Machine translation using deep learning: An overview, in: 2017 International Conference on Computer, Communications and Electronics (Comptelix), 2017, pp. 162–167. doi:10.1109/COMPTELIX.2017.8003957.

[34] B. Robaidek, R. Koncel-Kedziorski, H. Hajishirzi, Data-driven methods for solving algebra word problems, CoRR abs/1804.10718 (2018). arXiv:1804.10718.
URL http://arxiv.org/abs/1804.10718

[35] A. Patel, S. Bhattamishra, N. Goyal, Are NLP models really able to solve simple math word problems?, CoRR abs/2103.07191 (2021). arXiv:2103.07191.
URL https://arxiv.org/abs/2103.07191

[36] Z. Wang, A. S. Lan, R. G. Baraniuk, Math word problem generation with mathematical consistency and problem context constraints, CoRR abs/2109.04546 (2021). arXiv:2109.04546.
URL https://arxiv.org/abs/2109.04546

[37] W. Ling, D. Yogatama, C. Dyer, P. Blunsom, Program induction by rationale generation: Learning to solve and explain algebraic word problems, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 158–167. doi:10.18653/v1/P17-1015.
URL https://aclanthology.org/P17-1015

[38] L. Wang, D. Zhang, J. Zhang, X. Xu, L. Gao, B. T. Dai, H. T. Shen, Template-based math word problem solvers with recursive neural networks, Proceedings of the AAAI Conference on Artificial Intelligence 33 (01) (2019) 7144–7151. doi:10.1609/aaai.v33i01.33017144.
URL https://ojs.aaai.org/index.php/AAAI/article/view/4697

[39] Q. Wu, Q. Zhang, Z. Wei, X. Huang, Math word problem solving with explicit numerical values, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 5859–5869. doi:10.18653/v1/2021.acl-long.455.
URL https://aclanthology.org/2021.acl-long.455

[40] Z. Liang, J. Zhang, J. Shao, X. Zhang, MWP-BERT: A strong baseline for math word problems, CoRR abs/2107.13435 (2021). arXiv:2107.13435.
URL https://arxiv.org/abs/2107.13435

[41] P. Mehta, P. Mishra, V. Athavale, M. Shrivastava, D. Sharma, Deep neural network based system for solving arithmetic word problems, in: Proceedings of the IJCNLP 2017, System Demonstrations, Association for Computational Linguistics, Tapei, Taiwan, 2017, pp. 65–68.
URL https://aclanthology.org/I17-3017

[42] D. Huang, S. Shi, C.-Y. Lin, J. Yin, W.-Y. Ma, How well do computers solve math word problems? large-scale dataset construction and evaluation, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 887–896. doi:10.18653/v1/P16-1084.
URL https://aclanthology.org/P16-1084

[43] Q. Zhou, D. Huang, Towards generating math word problems from equations and topics, in: Proceedings of the 12th International Conference on Natural Language Generation, Association for Computational Linguistics, Tokyo, Japan, 2019, pp. 494–503. doi:10.18653/v1/W19-8661.
URL https://aclanthology.org/W19-8661

[44] S. Upadhyay, M. Chang, Annotating derivations: A new evaluation strategy and dataset for algebra word problems, CoRR abs/1609.07197 (2016). arXiv:1609.07197.
URL http://arxiv.org/abs/1609.07197

[45] S. Upadhyay, M.-W. Chang, Draw: A challenging and diverse algebra word problem set, 2015.

[46] Y. Wang, X. Liu, S. Shi, Deep neural solver for math word problems, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 845–854. doi:10.18653/v1/D17-1088.
URL https://aclanthology.org/D17-1088

[47] T. Carneiro, R. V. Medeiros Da No´Brega, T. Nepomuceno, G.-B. Bian, V. H. C. De Albuquerque, P. P. R. Filho, Performance analysis of google colaboratory as a tool for accelerating deep learning applications, IEEE Access 6 (2018) 61677–61685. doi:10.1109/ACCESS.2018.2874767.

[48] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual

Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 2002, pp. 311–318. doi:10.3115/1073083.1073135.
URL https://aclanthology.org/P02-1040

[49] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 74–81.
URL https://aclanthology.org/W04-1013

Sandip Sarkar is working as an Assistant Professor in the Department of Computer Science and Application, Hijli College. He received his B.Tech in Computer Science and Engineering from Jalpaiguri Govt. Engineering College in 2011, followed by the completion of his M.E in Computer Science and Engineering from Jadavpur University in 2014. Currently, he is actively pursuing his Ph.D. in Computer Science and Engineering at the same university. His present research focus encompasses Question Answering, Information Retrieval, and Deep Learning.

Dipankar Das is an Assistant Professor in the Department of Computer Science and Engineering, Jadavpur University, India. He received Ph.D. and Master's degrees from the Department of Computer Science and Engineering, Jadavpur University in 2013 and 2009 respectively. He received Bachelor's degree in Computer Science and Engineering from West Bengal University of Technology in 2005. His research interests are in the area of Natural Language Processing, Emotion and Sentiment Analysis, Affect Computing, Information Extraction and Language Generation. He has more than 100 publications in top conferences and journals and has served as an author over 15 Book Chapters. He is a member of the IEEE, ACL, HUMAINE groups.

David Pinto obtained his PhD in artificial intelligence and pattern recognition at the Polytechnic University of Valencia, Spain in 2008. At present he is a full time professor at the Faculty of Computer Science of the Beneme´rita Universidad Auto´noma de Puebla (BUAP) leading the laboratory and PhD Program of Language & Knowledge Engineering. His areas of interest include clustering, information retrieval, crosslingual NLP tasks, computational linguistics, robotics, augmented reality, virtual reality, mobile devices and complexity theory. He has written more than 100 papers and has developed research projects, whose products have been registered in the Mexican Institute of Industrial Property. He has created various laboratories, being the last one, the Language & Knowledge Engineering Lab, visited by several national and international researchers for establishing collaboration in different topics of artificial intelligence.