ISSN 2007-9737

Math Word Problem Solving: Operator and Template Techniques with Multi-Head Attention

Sandip Sarkar[1], Dipankar Das[2],
Partha Pakray[3], David Eduardo Pinto-Avendaño[4]

[1] Hijli College, Kharagpur,
India

[2] Jadavpur University, Kolkata,
India

[3] National Institute of Technology Silchar, Silchar,
India

[4] Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
Mexico

{sandipsarkar.ju, dipankar.dipnil2005, davideduardopinto}@gmail.com,
partha@cse.nits.ac.in

Abstract. The present article introduces an extensive approach that effectively addresses math word problems by leveraging the benefits of operator and intermediate template techniques with multi-head attention. Our method identifies a vital relationship between the mathematical statements and their corresponding mathematical equations. We examine the intricacies associated with math word problems containing multiple unknown variables, as they pose unique challenges compared to those with a single unknown variable. Furthermore, we extensively analyze math word problems involving fundamental mathematical operations such as addition, subtraction, division, and multiplication. In our experimental setup, we employ a sophisticated mechanism that leverages multi-head attention and enables our model to selectively focus on different aspects of the input, allowing it to capture the most relevant and recent information necessary for solving the problems more accurately. Our aim is to create a system that assists non-native English students throughout their academic endeavors. Our system is specifically designed to support them in effectively solving a broad spectrum of mathematical word problems.

Keywords. Math word problems, operator-based techniques, template-based techniques, multi-head attention.

1 Introduction

Numerous approaches for developing a math solver that operates automatically have been suggested by various researchers since 1960s [28, 27, 6, 37, 35]. According to several researchers, the promising outcomes can be achieved while using only small data sets with low variability [10, 16, 31]. Several factors can cause students to struggle with math word problems, such as math anxiety, limited memory capacity, weak counting skills, language barriers, and a deficiency of problem-solving techniques [7, 8].

doi: 10.13053/CyS-27-4-4769

Table 1. Description of the dataset

Dataset	Problem Statement	Template	Equations
Dolphin T2 Final	The difference between 2 numbers is 34 . the larger number is 4 more than 3 times the smaller number. What ’re the numbers ?	m - a * n = b, m - n = c	x-y=34.0, x-4.0=3.0*y
DRAW-1K	How much 25 % solution must be added to 42 cc of pure water to make at most 20 % salt solution ?	a * m - b * m = b * c	0.01*25*x=0.01*20*(x+42)

With the advent of Natural Language Processing (NLP), the researchers have employed various computational methods to comprehend, evaluate, and resolve math problems expressed in natural languages [23, 22].

NLP algorithms have the ability to analyze the text-based information provided in a math word problem, extract pertinent details, and transform them into mathematical equations that can be solved.

This procedure encompasses various stages, such as syntactic and semantic analysis, context modeling, and reasoning. However, one of the major obstacles in utilizing NLP for solving math word problems is the variability in how the problems are formulated.

The same problem can be conveyed through numerous distinct expressions, which necessitates the algorithm’s resilience in coping with variations in syntax, grammar, and vocabulary.

Thus, in general, utilizing NLP for solving math word problems has the capability to enhance math education and broaden its accessibility to a broader population of students.

By automating the math problem-solving process, NLP algorithms can aid students in concentrating on comprehending the fundamental concepts and fostering problem-solving abilities that are crucial for achieving success in mathematics.

Our objective is to develop a system that aids non-native English students in their academic journey. Our system is designed to assist them in resolving a wide range of mathematical word problems.

We developed an experimental setup that uses a sophisticated mechanism called multi-head attention. This mechanism allows our model to focus on different parts of the problem and gather the most important information for accurate problem-solving.

We also analyzed math word problems involving basic math operations like addition, subtraction, division, and multiplication. By understanding these operations thoroughly, our model becomes better equipped to solve word problems that require these operations.

While applying our model on two datasets, DRAW-1K and Dolphin T2 Final, our method achieves outstanding results, surpassing other techniques in terms of BLEU scores.

Specifically, in the DRAW-1K dataset, our approach achieves the highest BLEU score of 0.42 when we encompass problems containing both single and multiple variables.

Similarly, for the problems containing single variable in the Dolphin T2 Final data set, our method attains the highest BLEU score of 0.73.

On the other hand, delving into specific mathematical operations, our approach achieves remarkable BLEU scores of 0.41 for addition and 0.44 for multiplication operations, respectively in the DRAW-1K data set.

Similarly, in the Dolphin T2 Final data set, our method excels with the highest BLEU score of 0.76 for subtraction operation. These compelling findings affirm the effectiveness and versatility of our proposed method in addressing math word problems with predetermined structures.

Table 2. Operation presented in DRAW-1k and Dolphin T2 Final Dataset

Operations	**Dataset**	
	DRAW	**Dolphin_T2_Final**
Addition(+)	1223	1354
Subtraction(-)	1422	978
Multiplication(*)	2112	1641
Division(/)	157	-

By demonstrating remarkable performance across various mathematical operations and different numbers of unknown variables, our approach presents a promising avenue for enhancing the capabilities of solving math problems.

The present work focuses on math word problems and examines them from various perspectives. More information regarding the analysis can be found in the following sections. The arrangement of our paper is as follows.

The review of previous literature is covered in Section 2. Section 3 of the paper provides an overview of the data set, while Section 4 goes into detail about the methodology utilized in the research.

Information regarding the training process is furnished in Section 5. Our observations and the limitations of our system are discussed in Section 6. Ultimately, we conclude our research in Section 7.

2 Related Work

If we go back a few years back, since 2012, deep learning has been applied to a variety of natural language processing tasks, such as Question Answering [25], text simplification [21], sentiment analysis [24], machine translation [26] as well as math word problem [20, 19, 34, 14].

The task of solving math word problems using natural language processing (NLP) techniques is a well-studied area in the field of artificial intelligence and machine learning. We can review the works into several sub-fields as follows.

2.1 Rule-Based Approach

In case of rule-based approaches, researchers have tried to address the math word problems by employing several simple rule based techniques. These approaches involve either solving the problem directly or using natural language processing techniques to translate it into equations.

A machine-guided solution for Mathematical Word Problems (MWP) was developed by Bussaba Amnueypornsakul and Suma Bhat [29]. Their method emphasizes the comprehension of the fundamental structure present in mathematical word problems [1, 3].

2.2 Statistical Approach

Overall, in this area, one of the prominent works [13] presents a novel approach that combines statistical methods with a tag-based logic representation to enhance the accuracy and flexibility of solving math word problems [11].

2.3 Tree-Based Approach

Past research has extensively explored the use of natural language processing (NLP) techniques, specifically tree-based approaches, to address mathematical word problems [32, 15, 2]. These techniques heavily rely on syntactic parsing to represent the problem's structure as a tree.

This tree representation is crucial for generating the necessary equation to effectively solve the problem. A common tree-based strategy is to use dependency parsing to find connections between words in a math word problem. This helps to construct a detailed dependency tree that captures the relationships between the words.

On the other hand, the researchers introduce a new neural model, GTS, for solving math word problems by directly predicting an expression tree [36]. Motivated by the mechanism of human based math problem-solving, the model incorporates top-down goal decomposition along with bottom-up sub-tree embedding to enable explicit information flow within the expression tree.

2.4 Deep-Learning Approach

In recent years, there has been a lot of interest in deep learning-based approaches to solve math word problems [17, 25, 17, 26]. These techniques make use of deep neural networks to automatically identify the connection between a problem's text and its corresponding solution.

Zhang presents a novel approach to solve math word problems using a neural network architecture [38]. The proposed technique involves transforming the problem statement into a graph structure, which is then converted into a tree using a specific algorithm. This tree representation is then used to train a neural network to predict the solution of the problem.

The work present in the article [33] introduces a technique based on deep learning to tackle math word problems. The technique involves utilizing a recurrent neural network (RNN) to process and represent the problem, and then using a multi-layer perceptron (MLP) to make predictions regarding the solution.

According to Saxena [4], a more understandable technique for solving math word problems is proposed that involves a formalism based on operations to represent the problem, which can be easily interpreted. Subsequently, a neural network is applied to make predictions regarding the solution.

Utilizing reinforcement learning (RL) to optimize the model for solving math word problems is another deep learning-based strategy. The RL-based method is trained to map the problem's text into its corresponding equation and then optimized the model by engaging a reward function that penalized incorrect answers.

3 Dataset

We have performed experiments using two datasets: the Dolphin T2 Final dataset and the DRAW-1K dataset. These datasets are further described in the following sections. Additionally, Table 1 presents an example from each dataset, giving an overview of their contents. Similarly Table 3 describe the statistics of the dataset.

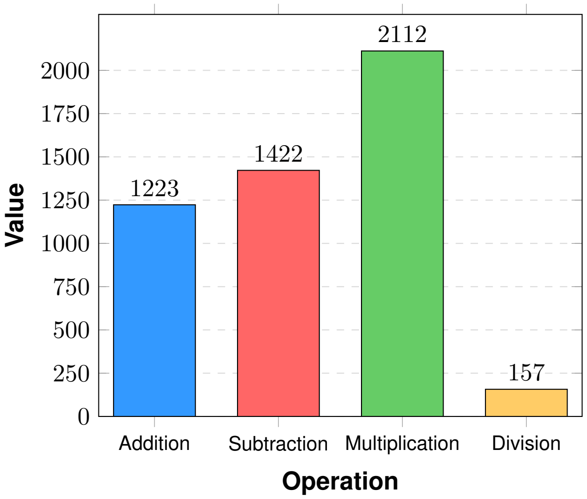


Fig. 1. Count of different Operations in DRAW-1K Dataset

3.1 DRAW-1K

The researchers in the study referenced as [30] examined more than 100,000 math problems[1]. These problem instances covered diverse mathematical concepts, addressing topics like quadratic equations and other non-linear equations. However, the main concentration of the study was directed toward the solution of algebraic word problems that specifically involved the application of linear equations systems.

In order to accomplish this goal, they employed keyword-matching techniques to filter out problems that involved non-linear equations.

Figure 1 displays various types of operations presented in the DRAW-1K dataset, while Figure 3 illustrates the top five templates along with their frequency in the DRAW-1K dataset. On the other hand, Figure 4 describes the histogram of the DRAW-1k dataset.

3.2 Dolphin T2 Final

The Dolphin T2 Final[2] is a subset of the Dolphin18K dataset that contains a wider range of problem types.

[1] algebra.com

[2] msropendata.com/datasets/f0e63bb3-717a-4a53-aa79-da339b0d7992

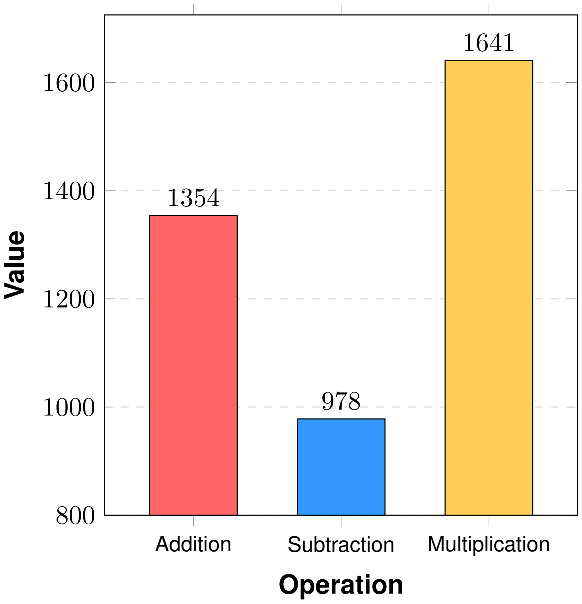


Fig. 2. Count of different Operations in Dolphin T2 Final Dataset

The main objective of the creators was to construct an extensive dataset comprising elementary mathematics problems [9]. They acquired the dataset from the mathematics section of the Yahoo! Answers website, which comprised a compilation of math problems accompanied by one or more associated solutions.

Figure 2 shows the different numbers of operation presented in Dolphin T2 Final Dataset. The Dolphin T2 Final dataset was generated with the intention of addressing mathematical word problems.

It encompasses a total of 831 problems that were originally posted by users on the community-driven question-and-answer platform, Yahoo! Answers [39]. Figure 3 exhibits the five most prevalent templates identified within the Dolphin T2 Final dataset.

Table 2 describes the observations that highlight the differences in the types and quantities of arithmetic operations present in the two datasets, indicating potential variations in the nature or purpose of the data captured in each dataset. On the other hand, Figure 4 describes the histogram of the Dolphin T2 Final dataset.

4 System Architecture

Our experiments employed a multi-head attention mechanism to address math word problems comprehensively. We approached the problems from different angles to gain a better understanding and improve the accuracy of our solutions.

1. Template-Based and Template-Independent Approach: We are dealing with math word problems with contain the template. So we take the math word problem with and without the template and see the performance of both approaches.

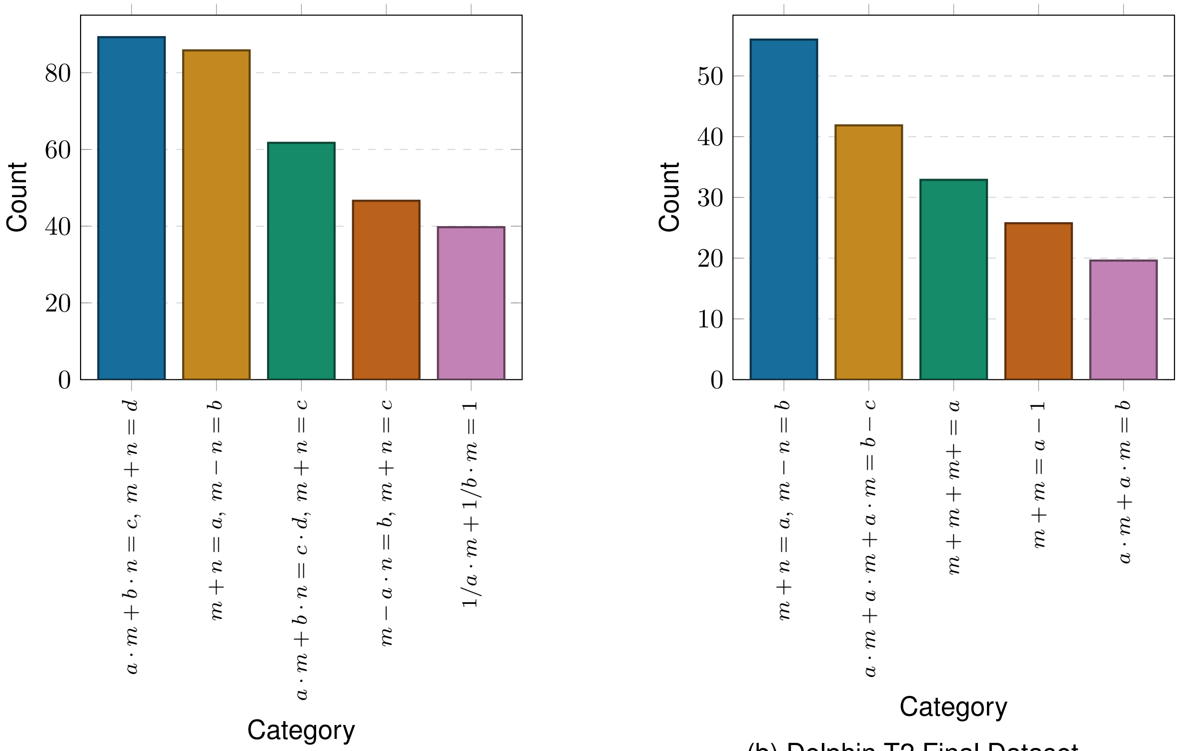
   In our first approach model can learn to focus on relevant information within the problem statement and extract key details to perform the necessary calculations. On the other hand, Math word problems that follow a specific template tend to have consistent structures, making them potentially easier to solve.

   By leveraging the multi-head attention mechanism, the model can learn to recognize and understand the template, which can help it quickly identify the necessary steps and operations required to solve the problem.

2. Unknown Variable Count-Based Approach: Math word problems contain different numbers of unknown variables. The number of unknown variables plays a vital role in the performance of our system. For this reason, we are dealing with that using two approaches 1) One unknown variable and 2) more than one unknown variable.

3. Top Frequent Template-based Approach: Our model benefited from the multi-head attention mechanism in recognizing common patterns within math word problem templates. Focusing on relevant parts of the problem statement that matched these patterns, it enhanced the efficiency of problem-solving.

4. Mathematical Operations based Approach: One effective approach is to partition the dataset based on these different operations. Subsequently, we can employ a multi-head attention mechanism to tackle the individual subsets of math problems.



(a) DRAW-1K Dataset

(b) Dolphin T2 Final Dataset

Fig. 3. Top 5 Template present in DRAW-1K and Dolphin T2 Final Dataset

Table 3. Statistics of Dataset

Dataset	Problems	Unique Equation	Unique Template	Percentage of Single Unknown Variables	Percentage of More than One Unknown Variable	Avg. Question Length	Avg. Equation Length
Dolphin T2 Final	831	770	215	26.23%	73.76%	75.64	12.93
DRAW-1K	1000	987	230	74.5%	25.5%	103.90	12.27

By dividing the dataset based on operations and incorporating a multi-head attention mechanism, we can effectively address math word problems.

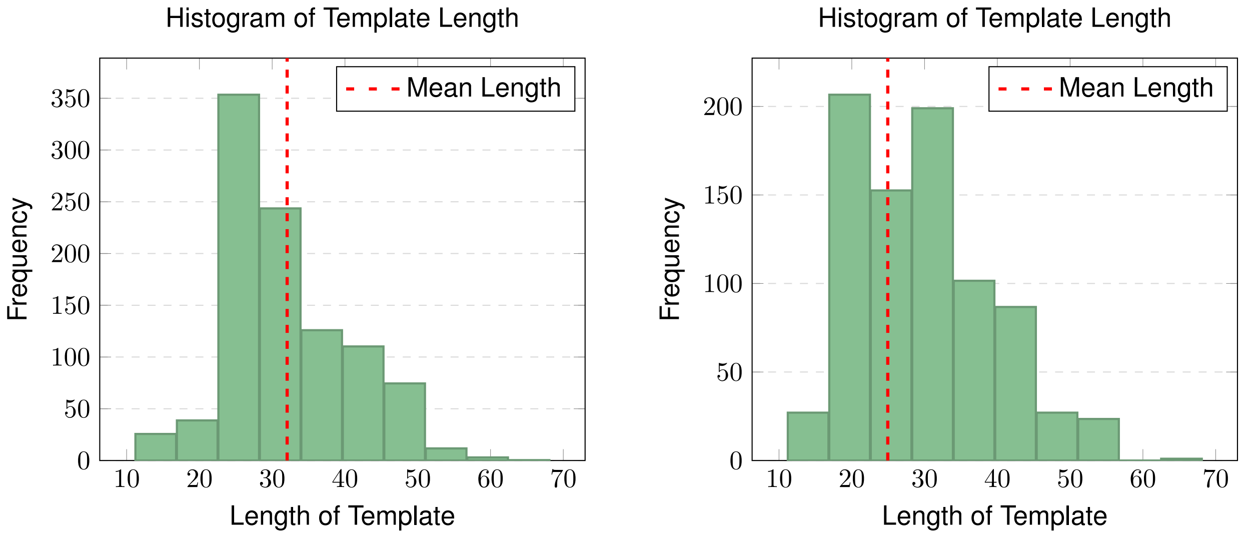
This strategy promotes specialization in individual operations and empowers the model to leverage its attention mechanism to better comprehend and solve math problems.

We used the Multi-Head Attention Mechanism which is a popular deep learning method extensively employed in different fields, including neural machine translation and natural language processing, with the aim of improving performance and attaining superior outcomes.

Figure 5 illustrates the architecture of the multi-head attention model utilized in our math word problem. As mentioned earlier, our experiments were carried out using two specific datasets, namely Dolphin_T2_Final and DRAW-1K.

The main objective of the Multi-Head Attention Mechanism is to boost the model's capacity to effectively focus on multiple important components within the input sequence at the same time.

This leads to an overall enhancement in performance. The multi-head attention mechanism partitions the input sequence into smaller “heads”, with each head concentrating on a distinct section of the input.



(a) Histogram of DRAW-1K Dataset

(b) Histogram of Dolphin T2 Final Dataset

Fig. 4. Histogram of DRAW-1K dataset and Dolphin T2 Final Dataset

For each head, the multi-head attention mechanism generates a query vector, key vector, and value vector based on the input sequence.

The query vector helps identify the important segments of the input sequence that require attention, while the key and value vectors calculate the weighted sum of the corresponding segments in the sequence.

The final output of the multi-head attention mechanism is obtained by passing the combined output from each head through a linear layer.

By allowing the model to simultaneously attend to multiple segments of the input sequence, this technique empowers it to detect intricate connections and dependencies that could be difficult to discern using a single attention mechanism.

5 Result

As mentioned earlier, our model is built upon a Multi-Head Mechanism and comprises four Encoder-Decoder layers with input/output embeddings of 128 dimensions. In our system, we integrated four self-attention heads with a batch size of 64.

We employed the Adam optimizer with a learning rate of 0.001 to optimize the models for DRAW-1K and Dolphin T2 Final dataset.

The training duration consisted of 30 epochs for DRAW-1K dataset and 55 epochs for Dolphin T2 Final dataset. The increasing popularity of cloud-based services can be attributed to their advantage of not necessitating any system setup or maintenance.

Various companies, including Amazon, Google, Azure, and Intel, offer cloud-based solutions. In our project, we made use of Google Colaboratory, a cloud-based service built on Jupyter Notebooks [5].

Jupyter is a user-friendly and open-source tool that can be employed both locally and in the cloud. It can be accessed through a web browser for convenience. However, Google Colaboratory has certain limitations, including the requirement to reconfigure settings after 12 hours of usage.

To assess the performance of our system, we employed two distinct evaluation metrics. The subsequent section offers a detailed explanation of these evaluation metrics.

BLEU: To assess machine translation quality, we employ the BLEU (bilingual evaluation understudy) metric.

Table 4. Template-Based and Template-Independent Based Result using Multi-head Attention Mechanism

				Template Independent Approach					
Dataset Name	**Dataset Variation**	**Count**	**BLEU**	**ROUGE-1**			**ROUGE-2**		
				r	**p**	**f**	**r**	**p**	**f**
DRAW 1K	**Single Variable**	255	0.22	0.5500	0.5535	0.5346	0.2485	0.2890	0.2588
	Multiple Variable	745	0.40	0.5700	0.5881	0.5760	0.4558	0.4358	0.4358
	Combined Dataset	1000	0.41	0.6198	0.6114	0.6029	0.4227	0.4187	0.4187
Dolphin_T2_Final	**Single Variable**	614	0.71	0.7738	0.7434	0.7562	0.6231	0.5734	0.5924
	Multiple Variable	217	0.51	0.6823	0.6724	0.6799	0.5430	0.5107	0.5244
	Combined Dataset	831	0.67	0.7577	0.7327	0.7425	0.6022	0.5701	0.5811
				Template Based Appraoch					
DRAW 1K	**Single Variable**	255	0.29	0.5721	0.5743	0.5771	0.2988	0.2986	0.2923
	Multiple Variable	745	0.41	0.5743	0.5928	0.5922	0.4681	0.4598	0.4624
	Combined Dataset	1000	**0.42**	**0.6258**	**0.6278**	**0.6177**	**0.4341**	**0.4295**	**0.4278**
Dolphin_T2_Final	**Single Variable**	614	**0.73**	**0.7738**	**0.7434**	**0.7562**	**0.6478**	**0.6552**	**0.6422**
	Multiple Variable	217	0.53	0.6913	0.6342	0.6489	0.5647	0.5356	0.5589
	Combined Dataset	831	0.68	0.7621	0.7581	0.7561	0.5852	0.5701	0.5811

This powerful metric compares the output of machine-translated text with one or more reference translations created by human translators.

By contrasting the machine translation against these human references, BLEU effectively evaluates the overall quality and accuracy of the translation output [18].

BLEU operates by counting the occurrences of shared n-grams, which are consecutive sequences of n words, in both the machine-generated output and the reference translations.

A higher score on the BLEU metric indicates a closer resemblance to human reference translations. Ranging from 0 to 1, the BLEU score is widely used in machine translation research to gauge the effectiveness of various models and techniques.

ROUGE: Text summarization output is evaluated for its effectiveness using a collection of metrics called ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [12].

The machine-generated summary is compared to the reference summaries created by humans by utilizing similarity metrics such as n-gram overlap and word order similarity.

The comparison entails quantifying the extent of overlap between the machine-generated summary and the reference summaries. ROUGE assigns scores ranging from 0 to 1, where higher scores indicate greater similarity between the machine-generated summary and the reference summaries.

ROUGE is widely used in the realm of natural language processing research to assess the efficacy of summarization models and algorithms.

As previously stated, we have thoroughly analyzed math word problems from various perspectives. Table 4 displays the outcomes obtained from both the Template-based and Template-independent approaches.

Furthermore, it demonstrates the influence of the number of unknown variables on the performance of the system.

We also examine the occurrence of comparable templates in the Math word problems. This analysis aims to identify similar problem types and assess the performance of our system on such cases. Therefore, we select the top 5 templates from each dataset.

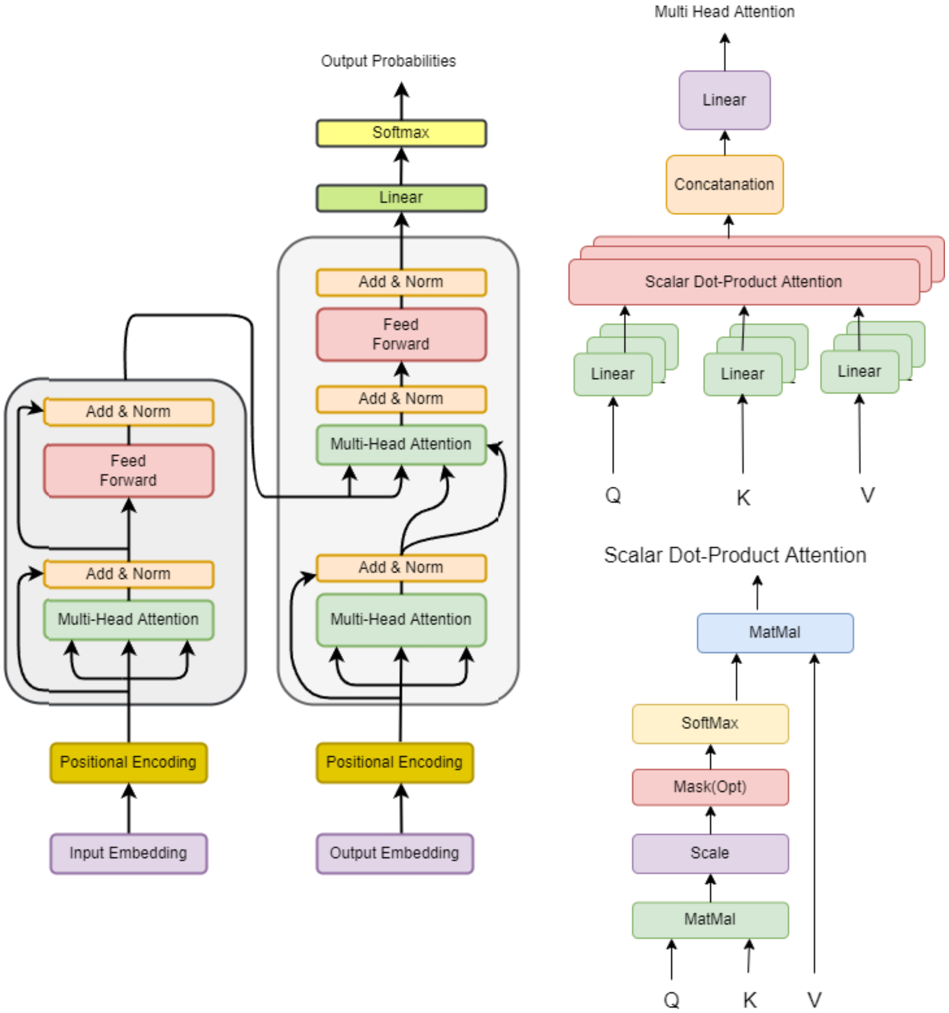


Fig. 5. Multi-Head Attention Model

The results of these five selected templates, as provided by our system, are presented in Table 5. Lastly, we assess the performance of the system from various primary mathematical operation perspectives. The outcomes of these datasets, categorized according to their respective mathematical operations, are presented in Table 6.

6 Observation

The research article presents an intriguing approach for effectively addressing math word problems with predetermined structures.

By leveraging the comprehensive datasets of DRAW-1K and Dolphin T2 Final, we establish a crucial correlation between the problem statements, templates, equations, and solutions. Through the integration of multiple datasets, a thorough examination of math word problems is made possible, covering a diverse set of mathematical operations and varying quantities of unknown variables.

In this section, we present the observation of our experiments on solving math word problems using the proposed system architecture and the multi-head attention mechanism.

Table 5. Result of Top Common Patterns in Template using Multi-head Attention Mechanism

Dataset Name	Top Template	Count	BLEU	ROUGE-1			ROUGE-2		
				r	p	f	r	p	f
DRAW 1K	a*m+b*n=c, m+n=d	88	0.42	0.6359	0.6342	0.6452	0.4246	0.4275	0.4285
	m + n = a, m - n = b	86	**0.43**	**0.6523**	**0.6421**	**0.6523**	**0.4352**	**0.4358**	**0.4481**
	a*m+b*n=c*d, m+n=c	62	0.36	0.6231	0.6231	0.6231	0.3981	0.3963	0.3979
	m-a*n=b, m+n=c	46	0.34	0.6048	0.6024	0.6077	0.3847	0.3845	0.3853
	1/ a * m + 1/ b * m = 1	42	0.40	0.6425	0.6490	0.6417	0.4052	0.4258	0.4288
Dolphin_T2_Final	m + n = a, m - n = b	56	**0.47**	**0.7256**	**0.7284**	**0.7278**	**0.4826**	**0.4853**	**0.4836**
	a*m+a*m+a*m=b-c	43	0.41	0.6825	0.6842	0.6835	0.4325	0.4356	0.4321
	m+m+m=a	34	0.43	0.7013	0.7025	0.7033	0.4526	0.4578	0.4612
	m+m=a-1	27	0.40	0.6671	0.6682	0.6623	0.4226	0.4378	0.4387
	a*m+a*m=b	17	0.38	0.6425	0.6490	0.6417	0.4052	0.4258	0.4288

We evaluate the performance of the model on different datasets, including variations with and without templates, as well as datasets categorized based on different mathematical operations. The subsequent sections provide further details regarding various forms of observations.

6.1 Template-Based and Template-Independent Approach

We observe that templates improved the model's performance across all metrics. For the DRAW 1K dataset, the model achieved a higher BLEU score of 0.29 for the single variable variation and 0.41 for the multiple variable variations. Similarly, for the Dolphin T2 Final dataset, the model achieved a higher BLEU score of 0.73 for the single variable variation and 0.53 for the multiple variable variations.

The ROUGE-1 and ROUGE-2 scores also demonstrated improvement for the datasets with templates compared to those without templates. These results suggest that the presence of templates aids the model in generating more accurate and contextually relevant solutions.

The Dolphin T2 Final dataset consistently outperforms the DRAW 1K dataset across both the template-independent and template-based approaches. This could be due to the Dolphin T2 Final dataset being more comprehensive, containing a larger variation count and potentially better-curated data.

6.2 Unknown Variable Count-Based Approach

Table 4 provide performance metrics for both single-variable and multiple-variable equation variations. Comparing the performance between these variations can provide insights into the model's ability to handle equations with different numbers of variables.

For Dolphin_ T2_Final dataset, the model may perform better on equations with a single variable, while in DRAW 1K dataset, it may show better performance on equations with multiple variables.

6.3 Mathematical Operations-based Approach

We further analyzed the performance of the model based on different mathematical operations. For the DRAW 1K dataset, the model achieved the highest BLEU score of 0.41 for the addition operation, followed by 0.34 for subtraction and division, and 0.40 for multiplication.

Similarly, for the Dolphin T2 Final dataset, the model achieved the highest BLEU score of 0.68 for addition, followed by 0.76 for subtraction, and 0.70 for multiplication.

These results indicate that the model performs well across different mathematical operations, with the subtraction operation showing the highest performance in terms of BLEU score for the Dolphin T2 Final dataset.

Table 6. Mathematical Operation Based Result using Multi-head Attention Mechanism

Dataset Name	Different Operations	Count	BLEU	ROUGE-1			ROUGE-2		
				r	p	f	r	p	f
DRAW 1K	**Addition**	780	**0.41**	**0.6039**	**0.6129**	**0.6084**	**0.4023**	**0.4025**	**0.4128**
	Subtraction	679	0.34	0.5577	0.5678	0.5624	0.3759	0.3685	0.3615
	Division	88	0.34	0.5679	0.5617	0.5673	0.3877	0.3812	0.3788
	Multiplication	901	0.40	0.6234	0.6345	0.6378	0.4572	0.4515	0.4652
Dolphin_T2_Final	**Addition**	743	0.68	0.7395	0.7345	0.7336	0.4884	0.4833	0.4828
	Subtraction	518	**0.76**	**0.7826**	**0.7853**	**0.7837**	**0.5262**	**0.5274**	**0.5268**
	Multiplication	647	0.70	0.7583	0.7571	0.7527	0.5014	0.5083	0.5076

6.4 Top Frequent Template in Template-based Approach

Lastly, we identified the top-performing templates based on the evaluation results. Table 5 showcases the top templates for the DRAW 1K and Dolphin T2 Final datasets, along with their respective counts and evaluation scores.

The number of occurrences for each template varies. The “m + n = a, m - n = b” template has the highest count in both datasets, with 86 occurrences in the DRAW 1K dataset and 56 occurrences in the Dolphin T2 Final dataset. The “m + n = a, m - n = b” template generally achieves higher scores compared to other templates, indicating better performance in terms of similarity to the reference outputs.

For the DRAW 1K dataset, the template “m + n = a, m - n = b” achieved the highest BLEU score of 0.43, followed by the template “a * m + b * n = c * d” with a BLEU score of 0.36. For the Dolphin T2 Final dataset, the template “m + n = a, m - n = b” achieved the highest BLEU score.

7 Conclusion and Future Work

Although significant advancements have been achieved in this field, there remain several unresolved challenges, such as addressing language and phrasing variations, tackling multi-step problems, and ensuring the accuracy of generated solutions.

In summary, the application of natural language processing (NLP) to solve math word problems has the capacity to greatly enhance mathematics education and make it more accessible for students of varying abilities. However, in order to fully realize its potential, further research and development are necessary.

Our proposition is a sequence-to-sequence (seq2seq) model that incorporates Multi-Head attention to generate equations from math word problems.

The experimental results of this approach on three widely utilized math word problem datasets substantiate its superior performance compared to the existing statistical model.

In the domain of math word problems, our model stands out as more advanced, particularly in handling a substantial quantity of unknown variables. While there are areas that can still be improved upon, we have the potential to enhance the accuracy of the system.

In the future, we intend to extend our work to generate nonlinear equations and apply these techniques to diverse word problem domains, such as physics, chemistry, and other related fields.

This paper has addressed the challenges posed by math word problems that involve more than two unknown variables and incorporates the four fundamental operations of addition, subtraction, division, and multiplication.

Future research in this field could focus on expanding the dataset and conducting further investigations into additional mathematical operations and problem structures. Through ongoing refinement and improvement of the proposed approach, there is the potential to bring about a revolution in math problem-solving.

This can significantly contribute to the development of intelligent systems that assist learners and educators in effectively tackling math word problems.

References

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

2. **Alvarez-Melis, D., Jaakkola, T. S. (2017).** Tree-structured decoding with doubly-recurrent neural networks. International Conference on Learning Representations.

3. **Amini, A., Gabriel, S., Lin, S., Koncel-Kedziorski, R., Choi, Y., Hajishirzi, H. (2019).** MathQA: Towards interpretable math word problem solving with operation-based formalisms. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1. DOI: 10.48550/arXiv.1905.13319.

4. **Amini, A., Gabriel, S., Lin, S., Koncel-Kedziorski, R., Choi, Y., Hajishirzi, H. (2019).** MathQA: Towards interpretable math word problem solving with operation-based formalisms. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Vol. 1, pp. 2357–2367. DOI: 10.18653/v1/N19-1245.

5. **Carneiro, T., Medeiros-Da-NóBrega, R. V., Nepomuceno, T., Bian, G. B., De-Albuquerque, V. H. C., Filho, P. P. R. (2018).** Performance analysis of google colaboratory as a tool for accelerating deep learning applications. IEEE Access, Vol. 6, pp. 61677–61685. DOI: 10.1109/ACCESS.2018.2874767.

6. **Griffith, K., Kalita, J. (2019).** Solving arithmetic word problems automatically using transformer and unambiguous representations. DOI: 10.48550/ARXIV.1912.00871.

7. **Hickendorff, M. (2021).** The demands of simple and complex arithmetic word problems on language and cognitive resources. Frontiers in Psychology, Vol. 12. DOI: 10.3389/fpsyg.2021.727761.

8. **Hong, Y., Li, Q., Ciao, D., Huang, S., Zhu, S. C. (2020).** Learning by fixing: Solving math word problems with weak supervision. DOI: 10.48550/ARXIV.2012.10582.

9. **Huang, D., Shi, S., Lin, C. Y., Yin, J., Ma, W. Y. (2016).** How well do computers solve math word problems? large-scale dataset construction and evaluation. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Vol. 1, pp. 887–896. DOI: 10.18653/v1/P16-1084.

10. **Ilany, B. S. (2010).** Language and mathematics: Bridging between natural language and mathematical language in solving problems in mathematics. Creative Education, Vol. 1, No. 1, pp. 138–148. DOI: 10.4236/ce.2010.13022.

11. **Liang, C. C., Wong, Y. S., Lin, Y. C., Su, K. Y. (2018).** A meaning-based statistical english math word problem solver. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, pp. 652–662. DOI: 10.18653/v1/N18-1060.

12. **Lin, C. Y. (2004).** ROUGE: A package for automatic evaluation of summaries. Text Summarization Branches Out, Association for Computational Linguistics, pp. 74–81.

13. **Lin, Y. C., Liang, C. C., Hsu, K. Y., Huang, C. T., Miao, S. Y., Ma, W. Y., Ku, L. W., Liau, C. J., Su, K. Y. (2015).** Designing a

tag-based statistical math word problem solver with reasoning and explanation. Proceedings of the 27th Conference on Computational Linguistics and Speech Processing, Vol. 20, pp. 58–63.

14. **Ling, W., Yogatama, D., Dyer, C., Blunsom, P. (2017).** Program induction by rationale generation: Learning to solve and explain algebraic word problems. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Vol. 1, pp. 158–167. DOI: 10.18653/v1/P17-1015.

15. **Liu, Q., Guan, W., Li, S., Kawahara, D. (2019).** Tree-structured decoding for solving math word problems. Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, pp. 2370–2379. DOI: 10.18653/v1/D19-1241.

16. **Mandal, S., Naskar, S. K. (2021).** Classifying and solving arithmetic math word problems—an intelligent math solver. IEEE Transactions on Learning Technologies, Vol. 14, No. 1, pp. 28–41. DOI: 10.1109/TLT.2021.3057805.

17. **Mehta, P., Mishra, P., Athavale, V., Shrivastava, M., Sharma, D. (2017).** Deep neural network based system for solving arithmetic word problems. Proceedings of the International Joint Conference on Natural Language Processing, System Demonstrations, Association for Computational Linguistics, pp. 65–68.

18. **Papineni, K., Roukos, S., Ward, T., Zhu, W. J. (2002).** Bleu: A method for automatic evaluation of machine translation. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 311–318. DOI: 10.3115/1073083.1073135.

19. **Patel, A., Bhattamishra, S., Goyal, N. (2021).** Are NLP models really able to solve simple math word problems?. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 2080–2094.

20. **Robaidek, B., Koncel-Kedziorski, R., Hajishirzi, H. (2018).** Data-driven methods for solving algebra word problems. .

21. **Sarkar, S., Das, D., Pakray, P., Pinto, D. (2021).** A hybrid sequential model for text simplification. Advances in Power Systems and Energy Management, pp. 33–42. DOI: 10.1007/978-981-15-7504-4_4.

22. **Sarkar, S., Das, D., Pakray, P., Pinto, D. (2022).** Formula retrieval using structural similarity. Conference and Labs of the Evaluation Forum, Vol. 3180.

23. **Sarkar, S., Das, D., Pakray, P., Pinto, D. (2022).** Generating equations from math word problem using deep learning approach. Computational Intelligence in Communications and Business Analytics, Springer International Publishing, pp. 252–259.

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

25. **Singh, D., Suraksha, K. R., Nirmala, S. J. (2021).** Question answering chatbot using deep learning with NLP. IEEE International Conference on Electronics, Computing and Communication Technologies, pp. 1–6. DOI: 10.1109/CONECCT52877.2021.9622709.

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

27. **Siyam, B., Saa, A. A., Alqaryouti, O., Shaalan, K. (2017).** Arabic arithmetic word

problems solver. Procedia Computer Science, Vol. 117, pp. 153–160. DOI: 10.1016/j.procs.2017.10.104.

28. **Sundaram, S. S., Khemani, D. (2015).** Natural language processing for solving simple word problems. Proceedings of the 12th International Conference on Natural Language Processing, NLP Association of India, pp. 394–402.

29. **Ughade, S., Kumbhar, S. (2019).** Survey on mathematical word problem solving using natural language processing. 1st International Conference on Innovations in Information and Communication Technology, pp. 1–5. DOI: 10.1109/ICIICT1.2019.8741437.

30. **Upadhyay, S., Chang, M. (2016).** Annotating derivations: A new evaluation strategy and dataset for algebra word problems. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Vol. 1, pp. 494–504.

31. **Verschaffel, L., Schukajlow, S., Star, J., Dooren, W. (2020).** Word problems in mathematics education: A survey. ZDM – Mathematics Education, Vol. 52, No. 1, pp. 1–16. DOI: 10.1007/s11858-020-01130-4.

32. **Wang, L., Wang, Y., Cai, D., Zhang, D., Liu, X. (2018).** Translating a math word problem to a expression tree. Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 1064–1069. DOI: 10.18653/v1/D18-1132.

33. **Wang, Y., Liu, X., Shi, S. (2017).** Deep neural solver for math word problems. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 845–854. DOI: 10.18653/v1/D17-1088.

34. **Wang, Z., Lan, A., Baraniuk, R. (2021).** Math word problem generation with mathematical consistency and problem context constraints. Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 5986–5999. DOI: 10.18653/v1/2021.emnlp-main.484.

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

36. **Xie, Z., Sun, S. (2019).** A goal-driven tree-structured neural model for math word problems. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, pp. 5299–5305. DOI: 10.24963/ijcai.2019/736.

37. **Yokoi, K., Aizawa, A. (2009).** An approach to similarity search for mathematical expressions using mathml. Towards a Digital Mathematics Library, pp. 27–35.

38. **Zhang, J., Wang, L., Lee, R. K. W., Bin, Y., Wang, Y., Shao, J., Lim, E. P. (2020).** Graph-to-tree learning for solving math word problems. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 3928–3937. DOI: 10.18653/v1/2020.acl-main.362.

39. **Zhou, Q., Huang, D. (2019).** Towards generating math word problems from equations and topics. Proceedings of the 12th International Conference on Natural Language Generation, Association for Computational Linguistics, pp. 494–503. DOI: 10.18653/v1/W19-8661.

Article received on 07/06/2023; accepted on 17/09/2023.
Corresponding author is Partha Pakray.