Journal of Theoretical and Applied Information Technology
31st July 2023. Vol.101. No 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MATH WORD PROBLEM SOLVERS: A REVIEW OF DATA DRIVEN APPROACHES AND DATASETS

HARSHAL KOTWAL[1,2], GIRISH KUMAR PATNAIK[3]

[1]Research Scholar, SSBT's College of Engineering & Technology, Jalgaon, India

[2]Department of Computer Engineering, SVKM's NMIMS, MPSTME, Shirpur, India

[3]SSBT's College of Engineering & Technology, Jalgaon, India

Email: [1]harshalrkotwal@gmail.com, [2]harshal.kotwal@nmims.edu, [3]patnaik.girish@gmail.com

ABSTRACT

Math Word Problem (MWP) solving is the process of finding the value of an unknown quantity in a word problem expressed in any Natural Language. The development of computer algorithms that can automatically solve arithmetic word problems is a challenge for the Artificial Intelligence research community. Solving Math Word Problems has recently become a significant research domain because automatically generating solution equations requires understanding natural language and its representation in computer understandable manner. This task is classified as natural language question-answering which is a well-studied problem in Natural Language Processing that requires machine common sense as well as domain knowledge and reasoning abilities. Solving a math word problem is particularly challenging due to the semantic gap in translating natural language text into machine-understandable logic that enables reasoning. To transform problems into several predefined templates in classification or retrieval style, early techniques depend on either predefined rules or statistical machine learning-based algorithms. Following the success of Machine Learning-based algorithms in a variety of domains, researchers have recently experimented with automatically solving Math Word Problems using large datasets. This research paper examines the performance and results of Data-Driven approaches to Solve Math Word Problems. The research also includes state-of-the-art small and large datasets used for performance evaluation of various approaches. The goal of this study is to help scholars understand the obstacles to solving Math Word Problems and to guide and motivate them to contribute in this direction.

Keywords: *Artificial Intelligence, Math Word Problems, Machine Learning, Natural Language Processing, Neural networks*

1. INTRODUCTION

Math Word Problems (MWPs) are text problems that need the use of mathematical formulae and some kind of mathematical logic based on specific facts to answer. MWPs are used to evaluate a student's logical reasoning and aptitude in a variety of subjects, including mechanics, physics, geometry, algebra, and others. An MWP requires the formulation of an equation to determine an unknown quantity using fundamental mathematical operations like addition, subtraction, multiplication, and division. Math Word Problems cannot be solved using pattern matching or end-to-end categorization methods. Translation of human-readable language into machine-readable logic structures is required for the method. As a result, developing intelligent systems capable of understanding and solving natural language MWP has been viewed as a crucial step toward general AI.

To find the unknown quantity, such a system must first understand the purpose of the question, as well as the semantic relevance of each known value or number in the issue text. Second, the system must use quantitative reasoning and knowledge of the subject area to produce the correct equation. Problems of varying degrees of complexity are insufficient. Following the availability of large MWP datasets, several recent attempts to apply deep learning to Solve Math Word Problems have been presented to study the prospects of Deep Learning-based techniques in solving Math Word Problems. Even with significant improvements in these methods, state-of-the-art techniques for large and diverse datasets containing a variety of Math Word Problems of differing levels of complexity are unsatisfactory.

According to Dongxiang Zhang et al. [1], Math Word Problems are divided into two types based on their features:

Journal of Theoretical and Applied Information Technology
31st July 2023. Vol.101. No 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- Arithmetic Word Problems: Arithmetic Word Problems are math problems with only one unknown, according to the definition. Based on the mathematical input stated in the problem, one or more known variables must be identified, and a mathematical equation must be generated to obtain the value of the unknown variable. Most of the time, these tasks are a one-step solution to a problem, meaning that their complexity is low. An example of an arithmetic word problem is illustrated in Fig. 1.
- Equation Set Math Word Problems: As there may be one or more unknown variables, Equation Set Problems are more challenging. Because the unknown variables are interdependent, determining the value of the desired unknown variable requires systematic equation construction. These problems are more complicated due to their multi-step nature, demanding the use of more advanced methodologies to solve them. Figure 2 illustrates an example of an arithmetic word problem.

Word Problem
A train 100 meters long completely crosses a 300 meters long bridge in 40 seconds. What is the speed of the train?
Equation
x = (100 + 300) / 40
Solution
x = 10 m/s

Figure 1: An example of Arithmetic Word Problem

Word Problem
A train 240 m long passed a pole in 24 sec. How long will it take to pass a platform 650 m long?
Equation
Speed = 240 / 24 = 10 m/s
Time Required = (240 + 650) / Speed = 890 / 10 = 89 seconds
Solution
x = 89 seconds

Figure 2: An example of Equation Set Math Word Problem.

1.1 Challenges in solving MWPs

AI researchers have long attempted to develop a computational system capable of replicating the human cognitive perspective to solve MWPs in many different ways. They have, however, only been able to solve a few simple word problems from basic school. Some MWP solutions are already available, however, an efficient MWP solver capable of tackling a large variety of problems remains unavailable. As an example, some basic low-complexity word problems are given below to help one understand the arithmetic MWPs that the researchers have been seeking to answer.

- There are 10 books in the drawer. Tim placed 20 books in the drawer. How many books are now there in total? Answer: 30
- The speed at which a man can row a boat in still water is 25 kmph. If he rows downstream, where the speed of the current is 11 kmph, what time will he take to cover 80 metres? Answer: 8 seconds
- If Tim had lunch at $50 and he gave 20% tip, how much did he spend? Answer: $60.00
- John found that the average of 15 numbers is 40. If 10 is added to each number then the mean of number is? Answer: 50
- A train covers a distance of 10km in 10 min. If it takes 6 sec to pass a telegraph post, then the length of the train is? Answer: 100 meter

The preceding word problems are extremely easy and straightforward for humans to solve. However, when it comes to computers automatically solving them from problem texts, even seemingly easy problems become challenging. Difficulties exist and may be encountered at almost every step. Some of the challenges are listed below.

- A variety of word issues with varying degrees of complexity
- Ambiguities caused by conjunctions and co-references
- Multiple verb/word sense
- Multi-word problem
- The presence of missing and unnecessary information in the problem phrases
- Inadequate or insufficient background information to answer the word issues

The accuracy/performance of the accompanying machine learning and/or natural language processing (NLP) tools/techniques employed.

The present review studies the performance of various Data Driven approaches to math word problem solvers in terms of their accuracy spread across diversified math word problem datasets from different domains. The review studies math word problem in English and Chinese Language only and does not consider math word problems in other languages.

1.2 Journey of MWP Solvers

Attempts to solve Math Word Problems automatically started during the 1960s, and the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

problem continues to pique the interest of academics since then. Many articles on this area have been published in the prime forums of artificial intelligence over the last several years. The problem becomes further more complex because of the challenges associated with the semantic gap in translating human-readable words into machine-understandable rules. We divide the evolution of MWP solvers into three key stages based on the technology underlying these solutions as given below.

1.2.1 Rule-based solvers:

Systems like STUDENT [2], DEDUCOM [3], WORDPRO [4], and ROBUST [5] used manually built rules and schemas for pattern matchings during the first pioneering period, roughly from 1960 to 2010. The performance of rule-based solvers depends on human input. In addition, these solvers can solve a very limited set of problems. Readers can refer to [6] for a comprehensive review of early rule-driven systems for automatic interpretation of natural language math problems as it has a full review of those early attempts at automatic understanding of natural language mathematics problems.

1.2.2 Feature Engineering based solvers:

MWP solvers in the second stage, used semantic parsing [7], [8], with the goal of mapping sentences from problem statements into structured logic representations to enable quantitative reasoning. It has regained significant academic interest, and several of those new approaches have been developed in recent years. For performance enhancement, these methods used a variety of feature engineering and statistical learning techniques. Semantic approaches learn how to convert problem text to a semantic representation that is transformed into an equation, by analysing data. Semantic representations use set-like constructions combined with hierarchical representations such as equation trees. Although such techniques are easy to interpret, no semantic representation that is flexible enough to handle all types of math word problems has been developed. We exclude them from this survey and focus on Data-driven solvers, which were not included in the previous surveys. Readers can refer to [9] for a review of Feature Engineering and Statistical Engineering based approaches.

1.2.3 Data Driven / Machine Learning based Solvers:

Deep learning's robustness has resulted in significant improvements in computer vision, speech recognition, language understanding, and other disciplines. If there is enough training data, Deep Learning can generate an adequate feature representation in a data-driven manner without the need for human interference. As a result, various attempts have been made to use Deep Learning to solve math word problems. These methods make use of deep learning models' capabilities as well as the availability of large training datasets. This is a new research direction for MWP solvers, and there have been multiple examples of Deep Learning-based algorithms being used to automatically answer Math Word Problems. As a result, we present a comprehensive review of data-driven MWP solutions that have recently been proposed.

If enough training data is available, data-driven models can be trained to convert math word problem texts into complex math equations. A further advantage of these models is that they do not rely on parsers or other NLP tools like semantic techniques do. Recently large size datasets for both English and Chinese word problems are introduced to train the algorithms. Experiments using classification, generation, and information retrieval models, along with notable variants of these models, show that a fine-tuned neural equation classifier often outperforms sophisticated solvers. We, therefore, propose a systematic study of data-driven approaches to solve math word problems and classify them as Retrieval based, Classification based, Generative, Ensemble and Hybrid models in the following sections.

The rest of the paper is structured as follows. In Section 2, we examine retrieval-based math word problem solvers, followed by classification-based MWP solvers in Section 3. In part 4, we examine generative models for MWPs. Sections 5 and 6 discuss ensemble and hybrid techniques respectively, to solve MWPs.

We also cover a detailed study of different datasets used for experiment purposes by various data-driven MWP solvers in Section 7. We provide the comparative performance of various MWPs on small and large size datasets in section 8. Finally, we conclude the study by highlighting various prospective directions in MWP solvers that need further investigation in the final section.

2. RETRIEVAL BASED MODELS

The retrieval-based models find similar questions from a dataset using a memory module and replace the values of the most similar problem with corresponding values in an unsolved problem. Authors A. Amini et al. [10] identified annotation

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

challenges concerning the current Math Word Problem solving datasets domain. According to the authors, existing datasets are either too small in scale or do not provide detailed and specific annotations for a wide variety of problem types.

The basic reason for this is that annotating math word problems properly across diverse problem categories is difficult even for humans and requires a background in arithmetic for annotators. The authors' contributions to the proposed research are three folded. First, they presented a large-scale collection of arithmetic word problems (MathQA). These problems are annotated with operation programmes aimed at improving the performance and applicability of the trained models. Second, the authors designed a new representation language for modelling operation programmes for each math problem. Finally, the researcher designed encoder-decoder neural models that translate word problems into a collection of possible operation programmes. The outcome of the operation programme is compared to a list of multiple-choice alternatives provided for a specific problem. The final model output is the matched solution.

The proposed method by K. Zaporojets et al. [11] uses tree-structured recursive neural network (Tree-RNN) configurations to investigate unique techniques for scoring candidate solution equations. This technique has the advantage of intuitively capturing the structure of the equations over more known sequential representations. Researchers looked at the reasoning component of answering arithmetic word problems. For the job of scoring candidate equations, they have proposed using two distinct Tree-LSTM topologies. The existing feature-based state-of-the-art model has a significant performance gap for mathematical problems with non-commutative numerous computations. This highlights the method's inability to reflect the intricacy of reasoning required to solve more difficult arithmetic problems.

3. CLASSIFICATION BASED MODELS

As per the authors, L. Wang et al., [12] LSTM based seq2seq model performs best to generate the math expression in the Math23K dataset. However, in a large space of target expressions, it suffers from performance degradation. For the production of math expressions, this research provides a template-based approach based on recursive neural networks. To anticipate a tree-structure template, the suggested model uses a seq2seq model. The leaf nodes in the tree are inferred numbers and unknown operators are inner nodes. The quantities are then encoded using a recursive neural network with Bi-LSTM and self-attention. The unknown operator nodes are then deduced in a bottom-up manner Authors proposed to use equation normalization and operator encapsulation to reduce the number of templates and improve the accuracy of template prediction.

MeSys is a meaning-based approach for answering English math word problems presented by the authors Chao-Chun Liang et al. [13]. The suggested system first analyses the text and then transforms both the body and question sections into their respective logic forms before inferring them. The relevant context information is expressed using proposed role tags to enable flexibility for annotating an extracted math quantity. The authors claim to have contributed by offering a new statistical model for selecting operands for arithmetic operations, which can be learnt automatically. A new informative and robust feature-set for selecting the desired arithmetic operation is also proposed. On common benchmark datasets given in the literature, the proposed approach outperforms other current systems significantly.

Y. Zou and W. Lu in [14] have introduced Text2Math, a unified structured prediction technique for solving both arithmetic word problems and equation parsing tasks. To automatically learn the relationship between words and arithmetic expressions, the proposed technique employs a novel joint representation. Unlike earlier efforts, Text2Math approaches the problem from end-to-end structured prediction, with the proposed algorithm aiming to predict the entire math statement as a tree structure at once.

Authors D. Lee and G. Gweon [15] have proposed a template-based – Multi-Task Deep Neural Network (T-MTDNN) framework that aims to address the challenges of understanding contextual relationships between multiple sentences and filling in missing background information in a Math word Problem. The T-MTDNN framework works in two steps. In the first step, the model generates a normalized-equation template (NET) using normalization. In the second stage, a best-matching template is chosen from a template list created in the previous step, utilising the MTDNN framework to solve a given arithmetic problem. The suggested method has two major benefits. First, knowing text representation is required to produce

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

equation templates that may be utilised as solutions. Text representation learning has been enhanced by using the MTDNN. Second, MTDNN improves the performance of solving arithmetic word problems by allowing domain adaptation with fewer in-domain labels.

Authors S. Mandal et al. [16] have proposed a Deep Learning based Arithmetic Word Problem Solver called DLAWPS. The proposed method comprises three modules. The first module accepts the input word problem which is sent through a Bi-directional Long Short-Term Memory based on a Recurrent Neural Network to predict the required operation among four fundamental operations $\{+, -, *, /\}$. The unnecessary quantities are eliminated in the second module, and the important quantities are identified using a knowledge-based irrelevant information removal unit (IIRU) to build an equation to solve arithmetic MWPs. Finally, the answer is generated by forming an equation from the identified relevant quantities. The proposed model is rule-based and problem-specific which can be scaled up with more rules for more problem types. The contribution stands out because of a template-free solution to generate the final answer.

To anticipate the proper equation set and ultimate result, authors A. Hevapathige et al. [17] have proposed a two-phase classifier, which is trained using templates for simultaneous linear equations. The first part determines the best appropriate template, which is then tested for the right answer using the available numerical values in the question in the next phase. Using this pipelining approach, the complexity of high dimensionality is reduced. In two phases, a novel system creates answers to math word problems. In the first phase, a template is projected for answering a question, and the second part involves the correct assignment of values.

Authors Rehman, Tayyeba, et al. in [18] proposed an MWP solver which is made up of three core modules: feature extraction, equation prediction, and equation solving. The proposed system uses NLP and a classification model to predict an equation template from a training dataset in the first step. In the next step, the system uses a reasoning module to instantiate the predicted template with nouns and numbers.

Due to a lack of training data and only one "standard" answer, the math word problem (MWP) is difficult. Authors Zhang, Jipeng, et al. [19] have offered Teacher-Student Networks with Multiple Decoders (TSN-MD), a unique approach that uses a teacher network to integrate the knowledge of equivalent solution expressions to better regularize the student network's learning behaviour. A teacher network and a student network make up the proposed TSN-MD. An encoder and two decoders are included in the student network, while the teacher network contains one more decoder than the other. This allows the student network to produce a wider range of mathematically comparable solution expressions than the ground truth.

Authors K. C. Win and N. L. Wah [20] have proposed an Automatic Question Answering System for generating an equation from the problem and forming a solution based on reasoning. Out of the two modules, the first module covers numerical queries with the CFG parser, whereas the second module covers word problems that the CFG parser cannot solve. The number quantifier and Unit Dependency Graph with Arithmetic Solver is used to solve the second section. The unit solving graph and the arithmetic solver are used to implement the system.

The MWPS tasks have primarily been investigated in the English and Chinese domains. However, investigations in less widely spoken languages, such as Korean, are rare. Authors K. Ki, D. Lee, and G. Gweon [21] executed a Korean MWPS task using the TAB (Template-based Arithmetic Solver with BERT) framework, which gives the state of the art performance on the MWPS task, to be able to undertake MWPS challenges in less popular languages.

Authors M. Shameem et al. in [22] suggested a hybrid system that includes an encoder-decoder model, a verb categorization model, and a regular expression model. Also, a new large mathematical word problem dataset with 57,400 questions and an accompanying expression template is constructed. The backbone of this hybrid mathematical solver constructed by Recurrent Neural Network is the encoder-decoder model (RNN). A verb categorization model and a regular expression model are integrated with an encoder-decoder model to improve accuracy.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4. GENERATION MODELS

According to W. Ling et al. [23], solving algebraic word problems requires a series of arithmetic operations to obtain a final answer, which they refer to as a program. The authors propose a technique for solving Math Word Problems by creating solution rationales, which are sequences of natural language and human-readable mathematical expressions that provide the final result. One of the three contributions of this research is the creation of a new dataset containing over 100,000 algebraic word problems with solutions and natural language answer rationales.

Second, they proposed a sequence-to-sequence model, which gives a set of instructions that, when executed, yields the rationale. The third contribution is a method for inferring programmes that provide reasoning and, eventually, an answer.

The paper by authors Ting-Rui Chiang et al. [24], presents a neural math solver based on an encoder-decoder framework to solve math word problems by operating symbols according to their semantic meanings in problem text. The suggested neural math solver, which mimics how humans think while writing equations, allows for a more accurate interpretation without the use of labelled rationales. The process of generating equations bridges the gap between the semantic world and the symbolic world. The encoder in the proposed model is aimed at understanding the semantics of problems, while the decoder works on tracking the semantic meanings of the generated symbols. After that, the model determines which symbol to create next. According to the authors, this method is the first to represent the semantic meanings of operands and operators for arithmetic word problems.

The Expression-Pointer Transformer (EPT) is a pure neural algebraic word problem solver presented by authors B. Kim et al. [25], generates solution equations for a Math Word Problem using the 'Expression' token and operand-context pointers. The authors claim to have contributed by providing a pure neural model (EPT) in the research paper. The EPT tackles the problem of expression fragmentation by creating 'Expression' tokens. Expression tokens then generate an operator and required operands at the same time. The EPT also solves the issue of operand-context separation by using operand-context pointers. The proposed model does not include any handcrafted features, although it achieves equivalent results to existing models that do.

Motivated by the success of deep Q-network in solving various problems with big search space and its promising performance in terms of accuracy and running time, authors L. Wang et al. [26], have proposed MathDQN as the first attempt of applying deep reinforcement learning to solve arithmetic word problems. The proposed automatic arithmetic word problem solver is a customized version of the general Deep Reinforcement Learning framework to fit the math problem scenario. The suggested approach consists of constructing the states, actions, and reward function, as well as a feed-forward neural network known as the deep Q-network. The suggested deep reinforcement learning-based model, according to the authors, achieves exceptional improvement on the majority of datasets.

According to the authors, Y. Shen and C. Jin in [27], most Seq2Seq algorithms for transforming text descriptions into equation expressions perform poorly due to insufficient consideration in encoder and decoder design. This is because these models only look at input/output entities as sequences, overlooking important structure information in text descriptions and equation expressions. By combining a sequence-based encoder with a graph-based encoder to improve the representation of text descriptions, the proposed model with multi-encoders and multi-decoders addresses these shortcomings. The model also creates various equation expressions using a sequence-based and a tree-based decoder. The framework of the proposed model is made up of four parts: a sequence-based encoder for obtaining the context representation of text descriptions; a graph-based encoder for integrating the dependency parse tree and numerical comparison information; a sequence-based decoder for generating the Abstract Syntax Tree's suffix order; and a tree-based decoder for generating the prefix order. Finally, the outcome is chosen based on the generation probabilities of various decoders.

As per the literature, Seq2Seq-based MWP solvers have the power of generating new expressions that do not exist in the training dataset. Another advantage of the Seq2Seq-based models does not rely on handcrafted features. Authors Z. Xie and S. Sun et al. in [28], however, claim that the Seq2Seq-based models do not match the goal-driven mechanism used by humans in problem-solving.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

When we read the problem text of an MWP, we usually try to figure out which target quantity is to be derived as the objective, and then we pay attention to the important information in the problem that can help us get there. If the objective can be achieved directly by using the relevant information, the problem is solved; otherwise, the process of deconstructing the goal into two sub-goals is repeated for each sub-goal until all goals are fulfilled. Motivated by the goal-driven process outlined above in human problem solving, the authors presented a novel neural model (dubbed GTS) to construct expression trees in a goal-driven way for arithmetic word problems. The proposed model first identifies and encodes the goal to achieve, which then gets decomposed into sub-goals combined by an operator in a top-down recursive way. The procedure is continued until the objective is simple enough to be calculated using a known quantity known as the leaf node.

Tree-based neural models for solving MWPs, which were recently proposed, have shown promising results in producing solution expression. Unfortunately, the majority of these models miss the connections and order information between the quantities. To address this issue, authors J. Zhang et al. [29], have proposed an MWP solver called Graph2Tree, which improves the performance by enriching the quantity representations in the MWP. The proposed Graph2Tree deep learning system combines the advantages of the graph-based encoder and the tree-based decoder to produce improved solution expressions. The Quantity Cell Graph and the Quantity Comparison Graph are the two graphs that make up the Graph2Tree system. The Quantity Cell Graph enhances the quantity representation by linking important descriptive terms in the problem text with the corresponding quantity. Similarly, the Quantity Comparison Graph is beneficial for preserving a quantity's numerical qualities while also utilizing heuristics to enhance representations of relationships between quantities.

The proposed system called MathBot in [30] learns to convert the English language-based math word problems into equations involving a few unknowns and arithmetic quantities. Authors A. K. Nayak et al., have built a Deep Neural Network architecture to make the MathBot learn to solve the equations. To improve its accuracy, researchers have introduced number mapping for word embedding as a novel technique to achieve high accuracy. Authors have also used an ingenious solution accuracy metric to evaluate the proposed models apart from the traditional BLEU score. The goal of the machine learning system would be to understand the arithmetic operation from the language context, identify the number of unknowns, extract the arithmetic quantities from the problem statement and construct the equation. Finally, the equation solver computes the final solution.

Authors K. Griffith and J. Kalita in their paper [31] describe the use of Transformer networks to convert math word problems to equivalent arithmetic expressions in infix, prefix, and postfix notations. In this study, the authors show that a Transformer based model can be used for machine translation from text to the language of arithmetic expressions. The proposed changes are well-motivated and easy to implement, demonstrating that the well-known Transformer language processing architecture can handle MWPs.

Authors S. Li et al. in [32], has introduced a revolutionary Graph-to-Tree Neural, which is a network that consists of a graph encoder and a tree decoder. This encodes a tree-structural input and decodes an augmented graph-structured output. The model learns how to map between one organised object, such as a graph, to another, such as a tree. The Graph2Tree model is flexible and independent of downstream tasks, allowing it to be used in a variety of NLP applications.

To make the initial attempt to represent the equations of multiple MWPs uniformly, authors J. Qin et al. in [33], suggested a basic method called Universal Expression Tree (UET), which is similar to the search tree of one-unknown linear word problems. They present a new dataset of Hybrid Math Word Problem (MWP) problems in this research. To improve the semantic representation of an issue, the authors suggested a subtree-level semantically aligned regularisation. They also presented a solver that can solve several types of MWPs with a single model in a unified manner.

The proposed study by authors Y. Meng and A. Rumshisky in [34], suggests using a Transformer-based methodology to build equations for arithmetic word problems. When copy and align techniques are not utilised, it beats RNN models and can even outperform sophisticated copy and align RNN models. The study also demonstrates that training a Transformer with two decoders, left-to-right and right-to-left, is effective in a generation task. Because of the ensemble effect, such a Transformer not only outperforms a single decoder

Transformer but also enhances the encoder training technique. The decoder in the original Transformer model is unidirectional, generating tokens from left to right based on previously created output. Authors have proposed using a Transformer to generate equations with two decoders operating in opposing directions based on this rationale. For the first time, a transformer is being used to construct mathematical equations.

According to authors D. Huang et al. [35], a sequence-to-sequence model has two flaws, first, it generates erroneous numbers, and second, it generates numbers in the incorrect locations. To solve these flaws, the authors have introduced a copy and alignment mechanism into the sequence-to-sequence model. Directly transferring the numbers from the problem description to the equations is referred to as copying. Alignment refers to the fact that the numbers in the equations and the numbers in the problem description are in sync. The alignment could be learned in a supervised manner by the model. Reinforcement learning is used to train the model which directly optimizes the solution accuracy.

The goal of the study proposed by D. Huang et. al. in [36] is to close the semantic gap between normal language and equations by presenting a new intermediate meaning representation technique for solving math issues. The proposed model modifies a sequence-to-sequence (seq2seq) network to generate the intermediate forms, based on recent work that seeks to construct equations from problem descriptions for this job. Previous research has demonstrated that seq2seq models can develop equations for problem categories that do not exist in the training data. Authors have offered a new method in this research that adds meaning representation and generates an intermediate form as an output. Furthermore, it is observed that the seq2seq model's attention weights repeatedly focus on integers in the problem description. The work also proposes using a type of attention regularisation to overcome the problem. The method proposes an iterative labelling method that uses signals from both equations and their solutions to train the model without explicit annotations of intermediate forms. The results of the experiments suggest that using intermediate forms for training outperforms directly mapping issues to equation systems.

Authors Kyung Seo Ki et. al. in [37] have proposed the GEO (Generation of Equations by Using Operators) model in this research, which avoids the usage of hand-crafted features and addresses two shortcomings that exist in existing neural models: 1. domain-specific knowledge characteristics are missing, and 2. encoder-level knowledge is being lost. The first instance of a missing domain-specific knowledge characteristic occurs when neural models do not make enough use of the domain-specific information in the target domain. Domain-specific knowledge is mathematical information in this situation of solving math word problems. People may recognize fundamental mathematical concepts such as numbers and operations by analyzing how humans collect and analyze mathematical information from a word problem while constructing an equation for the problem. On the other hand, using explicit knowledge of numbers and methods can help models perform better. The second issue of losing encoder-level information happens when decoder-learned hidden states are not utilized in the generation process in an encoder-decoder paradigm. The proposed GEO model may overcome the existing encoder-decoder based neural model's missing domain-specific knowledge features and loss of encoder-level knowledge in the domain of math word problem-solving.

Attempts to answer arithmetic word problems in the past have generated promising results, but they disregard background common-sense information that is not immediately provided by the problem. Furthermore, they focus on local features while ignoring global information during generation. To address these issues, authors Qinzhuo Wu et. al. [38] have presented a unique knowledge-aware sequence-to-tree (KA-S2T) technique for investigating how to better use external knowledge and capture global expression information. To capture common-sense information and facilitate word interaction, the proposed paradigm connects related items and categories based on external knowledge bases. To collect long-distance dependence and global expression information, the authors constructed a tree-structured decoder.

In this paper, authors P. Mishra et. al. [39] presented EquGener, a memory network-based equation generator with an equation decoder. A human math solver intuitively gathers crucial facts from the problem description that enable her to solve the problem. As a result of this, the authors created a deep semantic representation of the problem description that consists of many phrases that are conditional on the question at hand. The dense

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

representation learnt is utilised to decode an equation needed to answer the question. The authors describe a revolutionary word problem-solving architecture that integrates an end-to-end memory network with an equation decoder. Unlike most previous research, which can only handle a subset of operations, the proposed system can handle all forms of arithmetic operations.

The suggested method by authors Y. Hong et. al. [40] identifies three disadvantages of fully-supervised methods. First, the present MWP datasets only provide one solution for each problem, although other solutions exist that provide diverse approaches to solving the same problem. Second, annotating the expressions for MWPs takes time. Third, existing supervised learning techniques have a train-test disparity. To address these concerns, the authors suggest solving the MWPs with weak supervision. Learning with weak supervision naturally tackles the train-test mismatch by directly targeting answer accuracy rather than expression correctness. The proposed model is made up of a tree-structured neural model that generates the solution tree and a symbolic execution module that computes the answer. Authors also suggest a unique fixing mechanism to learn from inaccurate predictions to increase the efficiency of weakly-supervised learning. The error is propagated from the root node to the leaf nodes in the solution tree by the fixing function, which identifies the most likely fix that can give the desired answer.

Existing techniques for the MWP solution need complete supervision in the form of intermediate equations. Labelling each arithmetic word problem with its associated equations, on the other hand, is a time-consuming and expensive task. To address the problem of equation annotation, authors Chatterjee, Oishik, et al. in [41] have offered a unique two-step weakly-supervised approach for solving MWPs using only weak supervision. In the first phase, the proposed system learns to construct equations for questions in the training set using only the answer as supervision. The resulting equations and answers are then used to train any state-of-the-art supervised model in the second phase. Authors have additionally collected a new and rather large dataset in English with more than 10k instances, for training weakly supervised models for solving Math word problems.

Previous neural solutions for arithmetic word problems translated problem texts directly into equations without an explicit interpretation of the situations, and they frequently failed to handle more complex scenarios. Authors Hong, Yining, et al. [42] have offered the cognitive idea of a situation model called SMART, which is frequently used in psychology studies to model the mental states of people in problem-solving, to address such limitations of neural solvers. Problem-solving methods, like mathematics and logic, are thought to be used on the hallucinated situation model rather than the problem text. Authors have also created ASP6.6k, a new benchmark with four canonical types of algebra story problems including motion, price, relation and task.

The relevance of numerical values in problem-solving is overlooked by most methodologies, which consider numerical values in problems as number symbols. In the paper, authors Wu, Qinzhuo, et al. in [43], have presented NumS2T, a unique technique for effectively capturing numerical value information and utilizing numerical properties. The proposed model, in particular, uses a sequence-to-tree network with a digit-to-digit number encoder to capture number-aware problem representations and directly includes numerical values in the model. In addition, to make better use of the numerical properties, authors created a numerical properties prediction system. NumS2T predicts the comparative relationship between paired numerical values, specifies the category of each numeral, and quantifies its relevance to construct the final statement.

Authors Lee, Donggeon, et al. in [44] have presented a novel model called template-based multitask generation (TM-generation) in this paper that can increase the problem-solving accuracy of mathematical word problem-solving problems. A machine learning model should deduce an answer to a given problem by acquiring implied numeric information in an automatic mathematical word problem-solving task. To create an MWP solver that could perform potentially well for different types of problems it must address two challenges, first by filling in missing world knowledge required to solve the given mathematical word problem and second by comprehending the implied relationship between numbers and variables. Authors have proposed template-based multitask creation to address these two issues (TM-generation). To address the first challenge, the authors proposed the use of state-of-the-art ELECTRA models.

The paper published by Zhao, Wei, et al. [45] introduces Ape210K, a large-scale and diverse math word problem dataset containing 210K Chinese primary school-level questions and 56K different equation templates. Furthermore, a significant proportion of problems need the production of numbers that do not exist in the problem descriptions. All problems are collected from the online production system and are created by humans. Authors anticipate that Ape210K will help to push the bounds of automatically answering arithmetic word problems. Researchers also provide a copy-augmented and feature-enhanced seq2seq model on the Math23K dataset that outperforms the previous state-of-the-art system and serves as a strong baseline.

5. ENSEMBLE MODELS

Despite the simplicity of Sequence-to-sequence (SEQ2SEQ) models, there is still the issue that a math word problem can be solved correctly by more than one equation. The performance of maximum likelihood estimation is harmed by this non-deterministic translation. In this research, authors L. Wang et al. [46] have proposed an equation normalization approach to normalize duplicated equations, based on the uniqueness of expression trees. Experiments conducted to examine the effectiveness of three common SEQ2SEQ models in answering math word problems reveal that each model has its area of expertise when it comes to solving problems, hence an ensemble model is developed to combine their benefits. The research identifies two types of equation duplication including order duplication and bracket duplication. To normalize such order-duplicated templates, the proposed research work has presented two normalization rules. As per the first rule, two duplicated equation templates with unequal lengths should be normalized to the shorter one; and in rule two, the number tokens in equation templates should be ordered as close as possible to their order in number mapping.

6. HYBRID MODELS

In this study, authors Y. Wang et al. [47] suggested a Recurrent Neural Network model for solving an automatic math word problem. It is a sequence to sequence (seq2seq) model that converts natural language words into mathematical equations in math word problems. The authors created a similarity-based retrieval model and compared it to the seq2seq model that was proposed. Although seq2seq outperforms RNN on average, the retrieval model is capable of solving many problems that RNN fails to handle. The retrieval model's accuracy depends primarily on the similarity score between the unsolved problem and the problems in the training set. To improve the performance authors created a hybrid model that combines the seq2seq and retrieval models. The proposed hybrid model uses the retrieval model if the similarity score is greater than a certain threshold; or else, the seq2seq model is used to solve the problem.

7. MWP DATASETS

7.1 AI2:

The AI2 dataset contains 395 MWPs for third, fourth, and fifth graders that are single-step or multi-step arithmetic word problems, with 121 MWPs including irrelevant information. which can be solved with addition and subtraction. The dataset is taken from two websites: math-aids.com and ixl.com. It is then separated into three subsets as MA1, IXL, and MA2. IXL and MA2 are more difficult than MA1 because IXL has information gaps while MA2 contains irrelevant information in its math problems. Each math problem contains many quantities that may or may not be significant to the solution as per the observations by the authors in [1].

7.2 IL:

The problems were compiled from two websites k5learning.com and dadsworksheets.com. Problems that need previous knowledge (such as "apple is fruit" and "a week consists of seven days") are removed. The challenges are categorised based on textual similarities to enhance diversity. At most five problems are maintained for each cluster. Finally, 562 single-step word problems can be solved without the use of any unnecessary quantity using one of the four arithmetic operations i.e. (+, -, X, and /) as per the findings of the authors in the paper [1].

7.3 CC:

As per the authors in paper [1] the dataset is intended for multi-step mathematical problems. It was compiled from the commoncoresheets.com website and comprises 600 multi-step problems with no extraneous quantities. The dataset comprises different combinations of four fundamental operators including addition, subtraction, multiplication and division.

7.4 SingleEQ:

As proposed in the paper by authors in [35], the SingleEQ dataset given by Koncel-Kedziorski et al. is made up of 1,117 sentences and 15,292 words, as well as 508 arithmetic problems of varying difficulty (i.e., equations with single or multiple operators). Each word problem corresponds to a single valid equation with one unknown. The following operators are used in these equations: multiplication (*), division (/), subtraction (-), and addition (+). The information was acquired from the grade-school websites http://math-aids.com, http://k5learning.com, and http://ixl.com, as well as a subset of problems from Kushman et al.

7.5 AllArith:

Data from AI2, IL, CC, and SingleEQ are included in the dataset as stated by the authors in the paper [1]. All quantity references are converted to digit form. To measure how well an algorithm can distinguish between different problem types, duplicate problems are deleted. Therefore the problem size is reduced to 831. A dataset of 1,492-word math questions on a tiny size. The quantities in the text have previously been standardized into digit representation in this dataset.

7.6 Dolphin-S:

As per the observations of authors in paper [1], this dataset is a subset of Dolphin18K [12], which originally contained 18,460 questions and 5,871 equation-based templates. The dataset of Dolphin-S is made up of problems whose template is associated with only one problem. It contains 115 single-operator issues and 6,955 multiple-operator puzzles. Researchers have took a subset of 6,680-word math problems from Dolphin18k (Huang et al., 2016).

7.7 Math23K:

Math23K is a large-scale MWP dataset containing 23,162 Chinese math problems and math equation solutions for elementary school students gathered from various online education sources. The equation templates are extracted from 60,000 problems according to a set of rules [47]. A considerable proportion of problems that do not fit the guidelines are rejected to ensure high precision. Finally, there are 23,161 math problems with 2,187 templates. Every question has only one unknown variable and is linear. Math23K consists of problems with only one equation. There are problems with irrelevant quantities in the Math23K dataset. Other large-scale datasets with unlabeled problems or informal equation expressions include Dolphin18K (with 18,460 MWPs) and AQuA (with 100,000 MWPs). As a consequence, Math23K remains the most extensive and high-quality publication dataset on the market.

7.8 Alg514:

Problem posted by students on Algebra.com was crawled for the ALG514 MWP dataset. Problems containing information gaps are avoided. As a consequence, crowd workers on Amazon Mechanical Turk gathered and cleared a set of 1024 problems. Because the authors wanted each equation template to appear at least six times, these problems are further filtered. Finally, there are 514 problems left in the dataset. It has one equation as well as two simultaneous equation solved questions.

7.9 Dolphin1878:

The arithmetic problems in the Dolphin1878 MWP dataset were scraped from two websites: albegra.com and answers.yahoo.com as proposed in the paper by the authors in [1]. Human annotators manually add arithmetic equations and answers to math problems on answers.yahoo.com. Finally, the information from both sources is integrated to include 1,878 arithmetic problems and 1183 equation templates.

7.10 Dolphin18K:

Huang et al. developed Dophin18K. (2016). It has 5,738 templates and 18,460 math word problems from Yahoo!Answers. With 18,460 questions and 5,871 equation templates, the collected dataset is one of the largest thus far. The dataset is separated into two sections: the Dev set, which contains 3,728 problems, and the Eval set, which has 14,732 problems. With a lot of human participation, the problems, equation system annotations, and answers are retrieved semi-automatically.

7.11 DRAW1K:

According to the creators of DRAW1K, has limited textual variations and lacks structure. This prompted them to create a new dataset with a wide range of vocabularies and equation systems. With these two goals in mind, they created DRAW1K, an MWP dataset that crawls and filters 1,000 linear equation problems from algebra.com.

7.12 MAWPS:

Another testbed for arithmetic word problems with one unknown variable is MAWPS. The dataset is created from various websites with the purpose to include problems of different difficulty levels. The extracted dataset contains 2373 questions. MAWPS does not have information on grade levels.

7.13 MathQA:

The MATHQA dataset is a large-scale Math Word Problem dataset containing 37k English

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

pairings, where each math expression corresponds to an annotated formula for easier interpretation.

7.14 AQuA-RAT:

The AQuA offers over 100,000 GRE and GMAT-level arithmetic word problems. The problems are multiple-choice and cover a wide range of topics. Each question is divided into four parts: two inputs and two outputs: the description of the problem, marked as the question, and the potential (multiple choice) solution possibilities, designated as options. The purpose is to create a description of the rationale used to arrive at the correct answer, marked as rationale, as well as the correct choice label. The collection comprises a total of 104,519 problems, of which 34,202 are seed problems and 70,318 are crowdsourced problems as per the finding in the paper [1].

7.15 HWMP:

HMWP is a high-quality MWP dataset developed from a Chinese K12 math word problem bank, with each sample validating the universality of math word problem solvers and broadening MWP research frontiers to more closely reflect real-world scenarios. This set of problems contains arithmetic word problems, equation set questions, and non-linear equation problems. There are 5491 MWPs, including 2955 linear MWPs with one unknown variable, 1636 linear MWPs with two unknown variables, and 900 nonlinear MWPs with one unknown variable. The HMWP dataset is sufficient for proving the universality of math word problem answers because these problems cover the bulk of MWP scenarios. Math23K data is tagged with structured equations and answers.

7.16 NumWord:

There are 1,878 number word problems in Number Word Problem (NumWord). It has 986 questions in its linear subset (a subset of problems that can be solved using linear equation systems), all of which involve only four basic operations: +, -, *, and /.

7.17 ASDiv:

The ASDiv (Academia Sinica Diverse MWP Dataset) corpus is a new MWP dataset with a wide range of lexical patterns and problem types. Each problem has a set of consistent equations and solutions. It is labelled with the matching problem type and grade level, which can be used to assess a system's competence and to identify a problem's difficulty level. The various lexical patterns can be utilised to determine if an MWP solver gets answers by comprehending the meaning of the problem text or by just finding a similar MWP.

7.18 EMWP10K:

EMWP10K is a weakly supervised English MWP dataset of 10227 instances each associated with a single equation that may be used to train MWP solver models. The dataset was compiled by crawling the IXL website for arithmetic word problems ranging from grades VI to X. These word problems require a wide range of mathematical computations, from simple addition and subtraction to more difficult mensuration and probability calculations. There are ten different types of problems in the dataset, with three different levels of complexity.

7.19 Ape210K:

Ape210K is a large-scale, diverse math word problem dataset of 210K Chinese primary school-level problems and 56K unique equation templates. Furthermore, many problems require the creation of numbers that are not indicated in the problem descriptions. All problems are gathered using a human-created online production system. Ape210K can assist push the frontiers of automated arithmetic word problem-solving. Ape210K offers a variety of one-variable word problems in elementary school arithmetic. Professional math teachers write the problem-solving rationales and answers. Equations are taken from rationales to validate the equations with the solutions.

Table 1: Performance measure of MWP solvers on small datasets

				Dataset									
Approach	Author	Model	Year	MA1	IXL	AI2	IL	CC	SingleEQ	AllArith	Dolphin-S	ALG514	DRAW1K
Classification	[13]	MeSys	2018	-	-	69.8	70.6	-	-	-	-	-	-
	[14]	Text2Math	2019	-	-	86.5	81.4	-	-	-	-	-	-
	[17]	Two-Phase Classifier	2020	-	-	-	-	-	-	-	-	91.2	-
	[21]	KoTAB	2020	-	-	-	85.22	99.3	-	-	-	-	-
	[22]	Encoder-Decoder	2019	-	-	-	-	-	87.96	89.08	-	-	-
Generation	[24]	Encoder-Decoder	2018	-	-	-	-	-	-	-	-	28.86	-
	[25]	Transformer	2020	-	-	-	-	-	-	-	-	81.31	63.5
	[26]	MathDQN	2018	-	-	-	-	-	-	72.68	30.6	-	-
	[28]	GTS	2019	-	-	-	-	-	-	70.5	-	-	-
	[31]	Transformer	2019	-	-	77.2	96.4	94.4	-	-	-	-	-
	[33]	SAU-Solver	2020	-	-	-	-	-	-	-	-	57.9	-
	[35]	CASS	2018	-	-		-	-	-	-	-	82.5	-
	[37]	GEO	2020	-	-	-	-	-	-	-	-	82.1	-
	[39]	EquGener	2018	94.18	85.19	-	-	-	-	-	-	-	-
Hybrid	[47]	RNN-based seq2seq	2017	-	-	-	-	-	-	-	-	70.1	-
Retrieval	[11]	Tree-RNN	2021	-	-	-	-	-	75.47	-	-	-	-

Table.2. Performance measure of MWP solvers on large dataset

				Dataset							
Approach	Author	Model	Year	Math23K	Dolphin18K	MAPWS	MathQA	AQUA-RAT	HWMP	EMWP10K	Ape210K
Classification	[12]	seq2seq	2019	68.7	-	66.8	-	-	-	-	-
	[15]	T-MTDNN	2020	72.6	-	78.88	-	-	-	-	-
	[17]	Two-Phase Classifier	2020	-	61.3	-	-	-	-	-	-
	[19]	TSN-MD	2020	77.4	-	84.4	-	-	-	-	-
Ensemble	[46]	Equation Normalization	2018	68.4	-	69.2	-	-	-	-	56.6
Generation	[24]	Encoder-Decoder	2018	65.8	9.8		-	-	27.4	-	52.28
	[25]	Transformer	2020	-	-	84.51	-	-	-	-	-
	[23]	encoder-decoder	2017	-	-	-	-	36.4	-	-	-
	[27]	Multi-E/D	2020	78.4	-	-	-	-	-	-	-
	[28]	GTS	2019	74.3	-	82.6	-	-	-	-	56.56
	[29]	Graph2Tree	2020	77.4	-	83.7	-	-	-	-	-
	[30]	Bi-LSTM	2019	-	-	-	-	-	84	-	-
	[32]	Graph2Tree	2020	-	-	78.8	69.65	-	-	-	-
	[33]	SAU-Solver	2020	74.84	11.41	-	-	-	44.83	-	-
	[34]	Transformer	2018	-	22.1	-	-	-	-	-	-
	[35]	CASS	2018	-	33.2	-	-	-	-	-	-
	[36]	Seq2Seq	2018	-	16.8	-	-	-	-	-	-
	[37]	GEO	2020	-	-	85.1	-	-	-	-	
	[38]	KAS2T	2020	76.3	-	-	-	-	-	-	68.7
	[41]	WARM	2020	54.3	-	-	-	-	-	86.2	
	[43]	NumS2T	2019	78.1	-	-	-	-	-	-	70.5
	[44]	TM-generation	2021	85.3	-	85.2	-	-	-	-	
	[45]	Bi-LSTM	2020	77.5	-	-	-	-	-	-	70.2
Hybrid	[47]	RNN-seq2seq	2017	64.7	5.9	59.5	-	-	23.02	-	-
Retrieval	[10]	Encoder-Decoder	2019	-	-	-	54.2	37.9	-	-	-

8. PERFORMANCE EVALUATION

We have tabulated the performance of various experimental results reported from previous

Journal of Theoretical and Applied Information Technology
31st July 2023. Vol.101. No 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

works into two separate tables. The datasets have been classified into two classes based on the size of the datasets. Datasets with problem sizes close to 1000 are considered small datasets. In addition, the datasets with a problem size of more than 1000 are classified as large datasets. Such an integrated representation can help researchers in identifying the methods with higher performance in each dataset. Table 1 and Table 2 show the performance evaluation of different approaches on small and large size datasets respectively. Each row of both tables shows the approach being used and the column corresponds to the dataset used for the experiment. Each cell of the table gives information about performance in the percentage of the problem correctly solved by the algorithm used for solving the Math Word Problem. Cells with '-' value in Table 1 and Table 2, indicates that no experiment was conducted on the given dataset by the corresponding algorithm.

Most of the cells in Table 1 and Table 2 are marked with '-', meaning that an algorithm to solve MWPs was carried out on just one or two datasets. Moreover, the performance over small datasets is comparatively high as compared to the performance of data-driven approaches on large datasets. Solving Math Word Problems is therefore very challenging as there is no unique methodology that would work for different types of MWPs. In the following section, we discuss various challenges that are faced while addressing MWP.

9. CHALLENGES IN SOLVING MWPs

The following enumerates some of the challenges associated with the existing approaches.

- The mechanism of solving the word problems is considered the challenging factor due to the occurrence of the semantic gaps in parsing the natural-language content text to the understandable machine reading logic. This also enables the reasoning process.
- The efficiency of the solvers relies on handling the complexities in problem types of operations. The various word-solving problems also vary in the complexity degrees and the factors such as the ambiguities factors due to the co-reference and the conjunctions, Multiple words or multiple verbs, occurrences of the irrelevant information and the missing information in the problems sentences, temporal-ambiguities, insufficient background information necessary in solving the word-problems, Supporting-machine accuracy level and NLP-techniques.
- There exists the opportunity where the proposed aligning processes failed to perform well in larger size datasets and also in the diversified datasets.
- The prevailing of massive-features spaces modelled is the major challenge in the existing methodologies of mathematical-word problem-solving. This is due to the presence of some diverse templates.
- In considering the English-word problem-solving methods, involving AI-mechanism, the accuracy rate of the speech recognition and the language-translator process to the precise text in English is crucial to obtain the exact answers. Hence, It has been an inference that some of the faults are prone to occur due to language-translation inaccuracies. Further, it turns out to be vague to extract the knowledge and obtain the correct pattern.
- Some of the approaches follow the error-analysis phase, where similar problem types possess the various expressions of natural-language processing. Therefore future proceedings, ought to be focused on managing the challenge. Additionally, the expansion of the meaning-representation would assist in evolving more concepts of mathematics.
- The responses to the mathematical word problems exhibit the peculiar challenges in the logical-reasoning process upon the explicit quantities in text and the implicit quantities.
- A single problem of the arithmetic solving process comprises the narrative form and would ground the formalism of mathematical operation in the concepts of the real world. Hence establishing the problem-solvers would be a better challenge in cases of neural-network basis problem-solvers and human interventions. This also demands the logical reasoning of the solving-actions and the entities-relations.

10. CONCLUSION

This study elaborated on the various research studies that relied on the different kinds of Arithmetic English word problems, arithmetic mathematical word problems and the multi-lingual arithmetic word problems by the implementation of AI-artificial-intelligence in NLP-framework. These solvers utilized the various range of datasets suitable to evaluate the performance of the proposed systems. Further, the significant aspects of the components of the system are stated along with the advantages and disadvantages of the system also described. The solving process of the word problems found the applications used in many fields such as the E-learning-platform, Commerce-calculations, finance-domain calculations, Mathematical-question and answering systems and the process of mathematics-tutoring. The future relies on the implementation of MWP-framework handling the diversified, larger size of datasets. In the present decades, MWP-research studies obtained the goal-oriented and organized shape dependent on the standard-data-set types. The study would aid to get a clear picture of the present MWP-research status by various approach types, dataset types and component types.

REFERENCES:

[1] D. Zhang, L. Wang, L. Zhang, B. T. Dai, and H. T. Shen, "The gap of semantic parsing: A survey on automatic math word problem solvers," IEEE transactions on pattern analysis and machine intelligence, vol. 42, pp. 2287-2305, 2019.

[2] D. Bobrow, "Natural language input for a computer problem solving system," in Semantic information processing, M. Minsky, Ed. MIT Press, 1964, pp. 146–226.

[3] J. R. Slagle, "Experiments with a deductive question-answering program," Commun. ACM, vol. 8, no. 12, pp. 792–798, Dec. 1965.

[4] C. R. Fletcher, "Understanding and solving arithmetic word problems: A computer simulation," Behavior Research Methods, Instruments, & Computers, vol. 17, no. 5, pp. 565–571, Sep 1985.

[5] Y. Bakman, "Robust Understanding of Word Problems with Extraneous Information," ArXiv Mathematics e-prints, Jan. 2007.

[6] A. Mukherjee and U. Garain, "A review of methods for automatic understanding of natural language mathematical problems," Artif. Intell. Rev., vol. 29, no. 2, pp. 93–122, Apr. 2008.

[7] D. Goldwasser and D. Roth, "Learning from natural instructions," in IJCAI, 2011, pp. 1794–1800.

[8] T. Kwiatkowski, E. Choi, Y. Artzi, and L. S. Zettlemoyer, "Scaling semantic parsers with on-the-fly ontology matching," in EMNLP, 2013, pp. 1545–1556.

[9] Zhang, Dongxiang, et al. "The gap of semantic parsing: A survey on automatic math word problem solvers." IEEE transactions on pattern analysis and machine intelligence 42.9 (2019): 2287-2305.

[10] A. Amini, S. Gabriel, P. Lin, R. Koncel-Kedziorski, Y. Choi, and H. Hajishirzi, "Mathqa:

[11] Towards interpretable math word problem solving with operation-based formalisms," arXiv preprint arXiv:1905.13319, 2019.

[12] K. Zaporojets, G. Bekoulis, J. Deleu, T. Demeester, and C. Develder, "Solving Arithmetic Word Problems by Scoring Equations with Recursive Neural Networks," Expert Systems with Applications, vol. 174, p. 114704, 2021.

[13] L. Wang, D. Zhang, J. Zhang, X. Xu, L. Gao, B. T. Dai, et al., "Template-based math word problem solvers with recursive neural networks," in Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 7144-7151.

[14] C.-C. Liang, Y.-S. Wong, Y.-C. Lin, and K.-Y. Su, "A meaning-based statistical english math word problem solver," arXiv preprint arXiv:1803.06064, 2018.

[15] Y. Zou and W. Lu, "Text2math: End-to-end parsing text into math expressions," arXiv preprint arXiv:1910.06571, 2019.

[16] D. Lee and G. Gweon, "Solving Arithmetic Word Problems with a Template based Multi-Task Deep Neural Network (T-MTDNN)," in 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), 2020, pp. 271-274.

[17] S. Mandal, S. Ahmed, and S. K. Naskar, "Solving arithmetic word problems: A deep learning based approach," Journal of Intelligent & Fuzzy Systems, pp. 1-11, 2020.

[18] A. Hevapathige, D. Wellappili, G. U. Kankanamge, N. Dewappriya, and S. Ranathunga, "A two-phase classifier for automatic answer generation for math word problems," in 2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer), 2018, pp. 1-6.

[19] Rehman, Tayyeba, et al. "Automatically solving two-variable linear algebraic word problems using text mining." Expert Systems 36.2 (2019): e12358.

[20] Zhang, Jipeng, et al. "Teacher-student networks with multiple decoders for solving math word problem." (2020): 4011.

[21] K. C. Win and N. L. Wah, "Development of an Automatic Answering System to Arithmetic Questions," in 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), 2018, pp. 281-285.

[22] K. Ki, D. Lee, and G. Gweon, "KoTAB: Korean Template-Based Arithmetic Solver with BERT," in 2020 IEEE International Conference

on Big Data and Smart Computing (BigComp), 2020, pp. 279-282.

[23] M. Shameem, "Automatic Hybrid Mathematical Word Problem Solver," in proceedings of the International Conference on Systems, Energy & Environment (ICSEE), 2019.

[24] W. Ling, D. Yogatama, C. Dyer, and P. Blunsom, "Program induction by rationale generation: Learning to solve and explain algebraic word problems," arXiv preprint arXiv:1705.04146, 2017.

[25] T.-R. Chiang and Y.-N. Chen, "Semantically-aligned equation generation for solving and reasoning math word problems," arXiv preprint arXiv:1811.00720, 2018.

[26] B. Kim, K. S. Ki, D. Lee, and G. Gweon, "Point to the Expression: Solving Algebraic Word Problems Using the Expression-Pointer Transformer Model," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 3768-3779.

[27] L. Wang, D. Zhang, L. Gao, J. Song, L. Guo, and H. T. Shen, "Mathdqn: Solving arithmetic word problems via deep reinforcement learning," in Proceedings of the AAAI Conference on Artificial Intelligence, 2018.

[28] Y. Shen and C. Jin, "Solving Math Word Problems with Multi-Encoders and Multi-Decoders," in Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 2924-2934.

[29] Z. Xie and S. Sun, "A Goal-Driven Tree-Structured Neural Model for Math Word Problems," in IJCAI, 2019, pp. 5299-5305.

[30] J. Zhang, L. Wang, R. K.-W. Lee, Y. Bin, Y. Wang, J. Shao, et al., "Graph-to-tree learning for solving math word problems," 2020.

[31] A. K. Nayak, R. Patwari, and V. Subramanian, "MathBot–A Deep Learning based Elementary School Math Word Problem Solver."

[32] K. Griffith and J. Kalita, "Solving arithmetic word problems automatically using transformer and unambiguous representations," in 2019 International Conference on Computational Science and Computational Intelligence (CSCI), 2019, pp. 526-532.

[33] S. Li, L. Wu, S. Feng, F. Xu, F. Xu, and S. Zhong, "Graph-to-Tree Neural Networks for Learning Structured Input-Output Translation with Applications to Semantic Parsing and Math Word Problem," arXiv preprint arXiv:2004.13781, 2020.

[34] J. Qin, L. Lin, X. Liang, R. Zhang, and L. Lin, "Semantically-Aligned Universal Tree-Structured Solver for Math Word Problems," arXiv preprint arXiv:2010.06823, 2020.

[35] Y. Meng and A. Rumshisky, "Solving math word problems with double-decoder transformer," arXiv preprint arXiv:1908.10924, 2019.

[36] D. Huang, J. Liu, C.-Y. Lin, and J. Yin, "Neural math word problem solver with reinforcement learning," in Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 213-223.

[37] D. Huang, J.-G. Yao, C.-Y. Lin, Q. Zhou, and J. Yin, "Using intermediate representations to solve math word problems," in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 419-428.

[38] K. S. Ki, D. Lee, B. Kim, and G. Gweon, "Generating Equation by Utilizing Operators: GEO model," in Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 426-436.

[39] Q. Wu, Q. Zhang, J. Fu, and X.-J. Huang, "A Knowledge-Aware Sequence-to-Tree Network for Math Word Problem Solving," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 7137-7146.

[40] P. Mishra, L. J. Kurisinkel, D. M. Sharma, and V. Varma, "EquGener: A Reasoning Network for Word Problem Solving by Generating Arithmetic Equations," in PACLIC, 2018.

[41] Y. Hong, Q. Li, D. Ciao, S. Haung, and S.-C. Zhu, "Learning by Fixing: Solving Math Word Problems with Weak Supervision," arXiv preprint arXiv:2012.10582, 2020.

[42] Chatterjee, Oishik, et al. "A weakly supervised model for solving math word problems." arXiv preprint arXiv:2104.06722 (2021).

[43] Hong, Yining, et al. "Smart: A situation model for algebra story problems via attributed grammar." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 35. No. 14. 2021.

[44] Wu, Qinzhuo, et al. "Math Word Problem Solving with Explicit Numerical Values." structure 2018 (2017): 2019.

[45] Lee, Donggeon, et al. "TM-generation model: a template-based method for automatically solving mathematical word problems." The Journal of Supercomputing (2021): 1-17.

[46] Zhao, Wei, et al. "Ape210K: A Large-Scale and Template-Rich Dataset of Math Word Problems." arXiv preprint arXiv:2009.11506 (2020).

[47] L. Wang, Y. Wang, D. Cai, D. Zhang, and X. Liu, "Translating a math word problem to an expression tree," arXiv preprint arXiv:1811.05632, 2018.
[48] Y. Wang, X. Liu, and S. Shi, "Deep neural solver for math word problems," in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 845-854.