Journal of Interdisciplinary Mathematics
ISSN: 0972-0502 (Print), ISSN: 2169-012X (Online)
Vol. 28 (2025), No. 2, pp. 607–625
DOI : 10.47974/JIM-2105

Decoding math : A review of datasets shaping AI-driven mathematical reasoning

Michael Kurisappan *
S. Sundara Pandiyan †
Department of Computer Science and Engineering
Christ University
Bangalore 560074
Karnataka
India

Abstract

Math problem solving is a fundamental part of the modern era, and artificial intelligence (AI) driven mathematical reasoning has become an essential part of data work. In this literature review, we explore the diverse array of datasets intended to improve AI models' capacity to solve mathematical word problems. These datasets not only provided diverse problem sets but also served as benchmarks for evaluating the performance of various deep learning models, including recurrent neural networks (RNNs) and graph-based models. The datasets, particularly GSM8K, posed challenges that even the most sophisticated transformer models struggled to overcome, setting a new standard for the study of AI systems in math problem solving. This literature review aims to provide a comprehensive overview of the evolving landscape of mathematical problem solving, paving the way for future advances in AI-driven mathematical reasoning.

Subject Classification: *Primary 00A05, Secondary 97F90.*

Keywords: *Mathematical reasoning, Deep learning, Natural language processing, NLP datasets, Question answering, Semantic parsing, Large language models, Attention mechanism.*

1. Introduction

The landscape of mathematical problem solving has witnessed significant evolution, marked by the emergence of diverse datasets catering to various aspects of this domain. From template and rule-based

* *E-mail:* **michael.k@res.christuniversity.in (Corresponding Author)**
† *E-mail:* **sundarapandiyan.s@christuniversity.in**

datasets focusing on algebraic expressions and geometry problems to first-generation deep learning datasets and, more recently, datasets tailored for large language models (LLMs), the field has seen a plethora of contributions. In this literature review, we explore the diverse array of datasets intended to improve artificial intelligence (AI) models' capacity to solve mathematical word problems in this review of the literature.

The initial datasets, such as ALG514 and DRAW, laid the foundation for exploring algebraic expressions and mathematical reasoning [1, 2]. Dolphin1878 introduced the concept of a new language, Dolphin, to bridge natural language text and mathematical expressions [3]. However, challenges persisted, including alignment ambiguities in Dolphin problems. The GEOS dataset delved into geometry problems, utilizing rule-based text parsing and diagram parsing [4]. Despite its advantages, GEOS faced the drawback of limited accessibility.

The evolution continued with datasets like DRAW1K, MAWPS, Math23K, and AQuA, each offering unique challenges and opportunities for deep learning algorithms [5, 6, 7]. These datasets not only provided diverse problem sets but also served as benchmarks for evaluating the performance of various deep learning models, including recurrent neural networks (RNNs) and graph-based models.

As the era of large language models dawned, MathQA, MATH, and GSM8K entered the stage, elevating the complexity of mathematical problem-solving tasks [8, 9]. These datasets, particularly GSM8K, posed challenges that even the most sophisticated transformer models struggled to overcome, setting a new standard for evaluating AI systems in math problem-solving. Domain-specific datasets like Geometry3K, GeoQA, and UniGeo focused on geometry problems, introducing interpretability and unified geometric transformer models to improve accuracy in text and diagram semantic parsing [10, 11, 12]. ScienceQA expanded the horizon by addressing science and math problems, emphasizing multi-hop question answering and the importance of external knowledge [13].

The advent of financial natural language processing (NLP) led to the creation of datasets like FinQA and TABMWP, catering to the intricacies of numerical reasoning in financial contexts [14, 15]. The Financial NLP datasets aim to bridge the gap between textual information and tabular data, showcasing the demand for diverse problem-solving capabilities.

A critical shift occurred with datasets focusing on core math theorems, exemplified by TheoremQA, recognizing the limitations of large language models in handling complex mathematical theorems [16]. This dataset challenges the current paradigm by presenting 800 questions covering 350 theorems from various domains. Deep learning techniques, from Seq2Seq and Seq2Tree models to Graph2Tree architectures, demonstrated advancements in solving math word problems. The introduction of pre-trained approaches with LLMs marked a paradigm shift, with models like GPT-4 showcasing remarkable performance in reasoning and problem solving [17].

As we delve into the intricacies of these datasets and methodologies, this literature review aims to provide a comprehensive overview of the evolving landscape of mathematical problem solving, paving the way for future advancements in AI-driven mathematical reasoning.

2. Background

The creation of word problem solving models in mathematics relies heavily on datasets because they provide the necessary elements for problem solving in real world contexts, training, evaluation, and innovation. These are essential tools for professionals, researchers, and developers who wish to enhance artificial intelligence's mathematical reasoning abilities. In this section, we'll talk about the extensive development of datasets from their inception to the most recent advancements in the field of mathematical word problem solving. Table 1 demonstrates how well-known datasets have evolved over time to address different mathematical reasoning objectives.

2.1 *Template and Rule based math problem solving*

The early developed dataset ALG514 is mostly deals with algebra-based math tasks. Questions from the Internet make up the total of 514 tasks in the dataset [1]. Another template-based dataset with 1000 algebraic annotation tasks is called DRAW [2]. It contained 232 math-related word problem templates. Using these tiny datasets presents a challenge for deep learning algorithms.

Dolphin1878's primary goal is to understand semantic parsing and reasoning for mathematical word problems [3]. To connect Natural language text and mathematical expressions, the author of this work developed a new language called Dolphin language. The questions come from math classes in elementary schools and are collected from the Internet. The word problems in the dataset provide clear explanations of the mathematical relationships found in the text. The fact that alignment

Table 1

MWP datasets: Domain-specific, not ideal for LLMs

Dataset	Domain	Tasks	Year	LLM Benchmark
ALG514	Algebra	514	2014	No
DRAW	Algebra	1000	2015	No
Dolphin1878	Arithmetic /Algebra	1878	2015	No
GEOS	Geometry	186	2015	No
Dolphin18K	Arithmetic /Algebra	18460	2016	No
DRAW1K	Algebra	1000	2016	No
MAWPS	Arithmetic	1987	2016	No
Math23K	Algebra	23162	2017	No
AQuA	Arithmetic /Algebra	101449	2017	No
MathQA	Arithmetic /Algebra	37259	2019	No
ASDiv	Arithmetic /Algebra	1238	2021	No
SVAMP	Arithmetic /Algebra	3138	2021	No

ambiguities arise in over 35% of Dolphin problems are identified as one of its main drawbacks.

GEOS is centered on using the Stanford Dependency Parser to solve geometry math problems using the concepts of rule-based text parsing and diagram parsing [4]. These questions originate from the SAT; there are 186 tasks in the dataset. The fact that the GEO is still not publicly accessible is one of the disadvantages.

Dolphin18K dataset is mainly concerned with solving linear and nonlinear mathematical problems. The dataset contains math questions that are sourced from the Internet and generally taught elementary and middle schools. Out of the total 18460 tasks in the dataset, there are 5871 templates [18]. The primary finding of the dataset is that its performance significantly deviates from that of its smaller, less varied original dataset Dolphin1878[3].

2.2 *Inception of Deep Learning Datasets: Unveiling the First Generation*

DRAW1K [5] is well known for human generated derivation annotations to solve algebra problems in the dataset. Its foundation is the requirement that word problems be used to create derivations and apply textual numerical values to the derived equation in order to complete the task. The dataset, comprising 1000 tasks in total, is sourced from the website "algebra.com". Because of the well-annotated structure created by humans, many Deep learning algorithms have experimented with DRAW1K to solve problems [19].

A dataset repository available online, MAWPS [20] offers the ability to create datasets according to specific criteria. Researchers can dynamically add new problems to the dataset in accordance with their training needs thanks to this dataset. Because of its simplicity, some well-known deep learning models can eliminate non-grammatical word problems, reduce template overlap, and improve classification model accuracy. As it was discovered to be the best test bed for their problem solving, multiple deep learning attempts were made against MAWPS [21, 22, 23].

It is widely known that the Chinese dataset Math23K [6] is used as a test bed for several deep learning algorithms [24, 25, 26]. Comprised of 23162 elementary school math problems, primarily in the algebraic domain. Math23K is designed with annotated word problems that include solution expressions and answers, much like MAWPS. The considerably greater average clause count and expression length of Math23K in comparison to MAWPS suggests that the Math23K dataset is more complex than MAWPS. Table 2 lists the most recent and sophisticated datasets, which are used as benchmarks for large language models.

Table 2

MWP datasets: Domain-specific, good for LLM Benchmarks

Dataset	Domain	Tasks	Year	LLM Benchmark
MATH	Math	12500	2021	Yes
GSM8K	Arithmetic/Algebra	8500	2021	Yes
Geometry3K	Geometry	3002	2021	Yes
GeoQA	Geometry	4998	2021	Yes
MathQA-Python	Arithmetic/Algebra	23914	2021	Yes
TAT-QA	Finance	16552	2021	Yes
UniGeo	Geometry	9543	2022	Yes
ScienceQA	Science	21208	2022	Yes
FinQA	Finance	8281	2022	Yes
TabMWP	Arithmetic/Algebra	133815	2022	Yes
MultiHiertt	stem	10440	2022	Yes
TheoremQA	STEM	800	2023	Yes

The AQuA [7] dataset, which is based on the GMAT and GRE, includes information and questions suitable for usage at the university level. There are 101449 tasks in the dataset that cover probability, polynomials, arithmetic, and geometry. 70,318 extra questions were added to the dataset by crowd workers, which aids in increasing the diversity of question creation. Some intriguing research has been done using the AQuA dataset [8, 27, 28].

2.3 *Benchmark Datasets for large Language Models*

MathQA [8] and MathQA-Python [29] datasets are viewed as an enhancement of AQuA [7] dataset which focused on mapping a neural model to operational programs [30]. Additionally, the questions are harder and more intricate, requiring a higher degree of mathematical reasoning [31]. With the help of MathQA, word problems reasoning in math have significantly improved [31, 32, 33, 34].

MATH [9] is a ground-breaking dataset that includes 12500 tasks gathered from different contests. The dataset includes methodical step-by-step solution documentation. These procedures can be used to train models to generate derivations and their justifications. Utilizes this dataset as a multiple large language Model benchmark due to its nature [17, 35, 36]. The MATH benchmark and the answer explanations for each MATH problem, which models can use to create in depth solutions of their own, are two advantages of the dataset that the community can use to assess an individual's aptitude for solving mathematical puzzles.

GSM8K [37] is intended to identify the present shortcomings in multi-step reasoning while using large language models for math problem solving. The dataset demonstrates that even the largest transformer models are unable to perform well against GSM8K, offering a solid contender and testing ground for future developments in the study of math word problems. This dataset is widely used as a reference by LLMs due to its clearly defined annotation structure [37, 38, 39, 40]. A self-thinking process and well-written, grade school math problems are two advantages of the dataset.

2.4 *Domain specific datasets for Deep Learning*

Geometry3K [10], GeoQA [11] and UniGeo [12] datasets are primarily focused on geometry with annotations. Geometry3K introduces an Interpretable Geometry Problem Solver (Inter-GPS) dedicated for geometry. This method's main advantage over the previous GEOS [4] research is that the dense annotation makes it easier to map the semantic gap between the visual content and text description in geometry problems. The author of UniGeo introduced a unified geometric transformer model, which is used to pretrain and predict the mathematical expressions in a given geometry problem. The primary use of these datasets is to significantly improve the accuracy of geometry problems where large language models are having trouble addressing the current difficulties with text and diagram semantic parsing, mapping, and correlation.

ScienceQA [13] employs a novel approach to solving science and math problems by providing 21K multi-hop question answers accompanied by thorough annotations detailing the explanations and solutions. This is due to the fact that accurate answers to the multi-model questions require both external knowledge and multi-hop question answering. the creation of SCIENCEQA, which provides explanations for solutions and annotations for lectures, thereby bridging a gap in the scientific datasets currently in use. It shows how Chain-of-Thoughts (CoT) [41] improves performance and reliability in few-shot and fine-tuning learning via explanation generation for large language models.

2.5 *Financial NLP datasets for Deep learning*

FinQA [14] was created in response to the rising demand for financial NLP. The meticulously annotated dataset contains in depth explanations of numerical reasoning techniques from professionals in the field of finance. The information is selected with care from the earnings reports of the S&P 500 companies.

TABMWP (Tabular Math Word Problems), a recently developed dataset featuring 38,431 open domain grade level problems [15]. These challenges necessitate mathematical reasoning involving both textual information and tabular data. TABMWP ensures that each question is conveyed through a combination of an image, semi structured text, and a structured table, all aligned with a specific tabular context. The questions are of two types: free text and multiple choice. Additionally, each problem includes annotations with gold solutions, providing insights into the multi-step reasoning process.

MULTIHIERTT [42], which was developed from large financial reports, has a number of distinctive features, such as documents with multiple tables and lengthy unstructured texts [42]. The dataset was used to determine how the tables should be arranged in a clear hierarchy. The level of complexity and mental work required to answer each question is significantly higher than what is found in the current benchmarks. Enormous annotations of reasoning processes and supporting information are provided, enabling comprehension of complex numerical reasoning. In addition, MT2Net is a novel QA model that includes reasoning modules and a facts retrieval mechanism. These modules enable the model to gather pertinent supporting information from the text and tables and use it to carry out a sub-sequent reasoning task.

2.6 *Math theorems specific dataset for Deep learning*

TheoremQA [16] was created to deal with the difficult core math theorems that large language models currently face in their problem solving. When it came to performance, Big Language Models beat math word problem solvers when they were using cutting edge datasets like GSM8K [37]. They are still falling behind in solving problems involving mathematical theorems, though. It is common to discover that the generated expressions and outputs are inaccurate when working with mathematical theorems. Experts in the field assembled this dataset, which consists of 800 superb questions covering 350 theorems from a variety of fields, including science, math, and finance.

3. Deep Learning Models for MWP

The development of deep learning techniques has fundamentally changed how intelligent systems are trained and developed across a broad spectrum of artificial intelligence applications. The primary benefit that deep learning techniques offer is the ability to learn on their own without requiring human assistance when training massive amounts of data This is also true when it comes to solving math word problems, where DL-based techniques significantly increase performance, particularly when it comes to equation accuracy and answer accuracy deduced from a math word problem. Figure 1 illustrate the evolution of Deep learning models to solve math problems.

3.1 *Seq2seq models for Math word problem solving*

Typically, a Seq2Seq model produces a token sequence as its output. It is frequently applied to tasks where the output may be a variable length sequence, such as text generation, summarization, and machine translation. A preliminary attempt at applying DL (Recurrent Neural networks) techniques to mathematical problem solving is the Deep Neural Solver

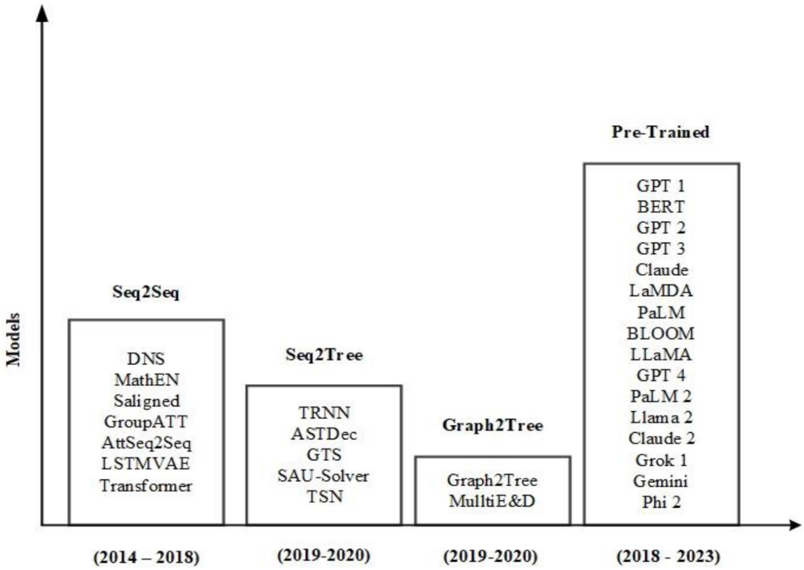

Figure 1

Evolution of Deep leaning models for Math Word problem solving

(DNS) [6]. In the hybrid DNS model, retrieval and seq2seq models are combined. The seq2seq model efficiently converts math word problems into equation templates by using gated recurrent units (GRU) [43] as an encoder and long short- memory (LSTM) [44] cells as a decoder. While addressing problems, the retrieval model is employed to provide even more performance enhancements. Later, more important works using seq2seq DL methods were conducted by researchers who increased the performance advantage by using LSTM and Bi-LSTM [45] as encoders and LSTM as decoders [46, 47, 48, 49].

3.2 *Seq2Tree models for Math word problem solving*

A tree structure is the result of a Seq2Tree model, as opposed to a Seq2Seq model. When creating or parsing syntactic structures, for example, this structure works especially well for tasks where the output can be arranged hierarchically. Numerous notable publications have employed Seq2Tree DL models and discovered a more advanced approach to solving math word problems with a high degree of accuracy when compared to Seq2Seq models [50, 51, 52, 53]. One notable contribution in this domain is the work by GTS [54], which employs a neural network with a tree structure to tackle mathematical problems. This model adopts a goal- driven methodology, in which the math problem is initially analyzed to identify various goals to be accomplished using the encoding technique based on GRU. Each goal is represented as a sub tree within a larger tree structure, and these are combined using operators to generate the final mathematical expression and corresponding answers. Apart from excelling all previous attempts, this model also makes sure that no erroneous math expressions or ambiguous numbers are produced during problem solving.

3.3 *Graph2Tree models for Math word problem solving*

While the Seq2Tree DL models outperform all other prior attempts at math problem solving, they still lag in terms of determining the relationship between statements and methodically organizing the quantities in each math problem. DL models based on Graph2Tree have been developed to tackle this issue. To solve MWP, Graph2Tree [55] introduces a graph-based encoder that is followed by a tree based decoder that creates expression trees The first step in this architecture is to determine the relationship between quantities and its attributes using a quantity cell graph and quantity comparison graph. This method resolves the relationship issue that the previous works raised. Second, the suggested graphs are used to teach a graph transformer the quantity relationships. The last step is to create the final math expression tree using a tree structure decoder and the values from the graph transformer. Graph2Tree method is also attempted with GRU+GCN [56] as an encoder and GRU as a decoder [24]. Experiments demonstrate that the Graph2Tree approach can perform better than earlier state-of-the-art solutions. Table 3 demonstrates the accuracy of equations and answers achieved by early DL models when tested against the Math23K and MAWPS datasets.

3.4 *Pre-trained approach for math problem solving*

For NLP, large language models (LLMs) represent the latest and most sophisticated technological revolution. With the increasing use of generative AI tools in the modern era, language learning and generations with LLM have become an essential part of data work. LLM’s are also called as foundation models [57] because of its vast capability to handle muti-tasking. Unlike the earlier versions of deep learning approach, LLM’s use a pre-trained leaning approach with massive amount of data. LLMs utilize billions of parameters in the pre-training phase, requiring a significant amount of processing power. Weights or variables are the parameters that a model learns during training. In light of user input, language models use these learned weights to learn how to anticipate the next potential word sequence and ultimately produce context aware text and other types of data. For best results, the model needed to be fine-tuned using task-specific data (small, labelled corpus) after the pre training phase.

4. Future Research Directions

Transformer [58] based artificial neural networks are the foundation upon which LLMs are built. The transformer mechanism can be better illustrated via the mathematical representation below. The self-attention mechanism in a Transformer model is defined by the attention scores and the weighted sum operation. Given a sequence of input vectors $x_1, x_2, \ldots, x_n$, the attention scores Attention (x_i, x_j) between positions i and j are calculated using a compatibility function and then used to weight the input vectors:

$$\text{Attention}(x_i, x_j) = \frac{\exp(\text{score}(x_i, x_j))}{\sum_{k=1}^{n} \exp(\text{score}(x_i, x_k))} \tag{1}$$

where score (x_i, x_j) is the compatibility score between xi and xj. In the Transformer architecture, layer normalization is often applied. The normalization process is mathematically represented as:

$$\text{Layer Norm}(x) = \gamma \cdot \frac{x-\mu}{\sigma} + \beta \tag{2}$$

where μ and σ are the mean and standard deviation of x, and γ and β are learnable parameters.

Loss function is an essential part of machine learning model training, which includes large language models. Its goal is to measure the discrepancy between the model's anticipated outputs and the actual target values found in the training set. Minimizing this loss function is the aim of training. For a sequence of words $w_1, w_2, \ldots, w_t$ in a sentence, and predicted probabilities $P(w_t \mid w_1, w_2, \ldots, w_{t-1})$, the cross-entropy loss is computed as:

$$\text{Entropy Loss} = -\frac{1}{T}\sum_{t=1}^{T} \log P(w_t \mid w_1, w_2, \ldots, w_{t-1}) \tag{3}$$

Table 3

Deep Learning for Single-Equation Math Word Problems (Math23K & MAWPS)

model	**Dataset**			
	math23k		mawps-single	
	Equ. Accuracy	Ans. Accuracy	Equ. Accuracy	Ans. Accuracy
DNS	57.1	67.5	78.9	86.3
MathEN	66.7	69.5	85.9	86.4
Saligned	59.1	69	86	86.3
GroupATT	56.7	66.6	84.7	85.3
AttSeq	57.1	68.7	79.4	87
LSTMVAE	59	70	79.8	88.2
Transformer	52.3	61.5	77.9	85.6
TRNN	65	68.1	86	86.5
AST-Dec	57.5	67.7	84.1	84.8
GTS	63.4	74.2	83.5	84.1
SAU-Solver	64.6	75.1	83.4	84
TSN	63.8	74.4	84	84.7
Graph2Tree	64.9	75.3	84.9	85.6
Multi E&D	65.5	76.5	83.2	84.1

The training objective for language modelling is often expressed in terms of the log likelihood of the training data. Given a sequence of words or tokens $w_1, w_2, \ldots, w_n$ in the training dataset, and the model's predicted probabilities $P(w_i \mid w_1, w_2, \ldots, w_{i-1}; \theta)$ for each word w_i in the sequence, where θ represents the model parameters, the training objective can be written as:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \log P(w_i \mid w_1, w_2, \ldots, w_{i-1}; \theta) \tag{4}$$

where N is the total number of words in the training dataset. - $\log P(w_i \mid w_1, w_2, \ldots, w_{i-1}; \theta)$ is the log-likelihood of predicting the i^{th} word given the context.

The most recent LLMs have multi model capability, which enables them to process both text and visual data. Some of the most well-known names in the field are Claude 2[59] from Anthropic, Google's PaLM (used

Table 4
LLM performance on MATH and GSM8K datasets

Type of Model	Model	GSM8k	MATH
Closed Source	Gemini Ultra	94.4	53.2
	GPT-4	92	42.5
	Claude 2	88	-
	Claude 1.3	85.2	-
	Flan-PaLM 2	84.7	33.2
	ChatGPT	80.8	34.1
	PaLM 2	80.7	34.3
Open Source	WizardMath	81.6	22.7
	LLaMA 2	56.8	13.5

in Bard) [60], LLaMA from Meta, and GPT-4 (used in chatgpt) from OpenAI [61].

4.1 *Progress with Large Language Models (LLMs) in Math Problem Solving*

With expression generation and answer accuracy exceeding 90%, GPT-4 [17], the most recent state-of-the-art multi model LLM, has made amazing progress in math word problem solving. GPT-4 perform better at solving math word problems than all previous state-of-the-art models. In some of the use cases, the performance solves mathematical reasoning better than human intelligence. Table 4 displays test results from a variety of recent benchmark datasets, including MATH and [9], GSM8K [37], for math problems that demonstrate the accuracy of reasoning and problem solving of popular LLM's for various benchmark Math datasets. According to Google's most recent multi-modal research model named as Gemini can perform better in math problem solving and reasoning than the state-of-the-art work gpt-4. The research community is eager to learn more about this rapidly developing field, and the future seems bright.

4.2 *Research progressive challenges with LLM's*

A chain-of-thought prompting method [62] is utilized by LLMS to facilitate intermediate reasoning skills for math problem solving. LLMs produce arithmetic errors, according to some recent research [9, 38, 63]. One of the most recent works [64] tackles this problem by solving equations using an external symbolic solver and LLM. Therefore, mathematical mistakes are prevented by utilizing an external symbolic solver capable of solving the equations. Research groups can no longer obtain the most popular LLMS for free for upcoming projects because it has been commercialized due to its high level of accuracy and widespread use. Recent research in the open-source space offers some hope for resolving this issue. For the Llama-2 Reinforcement Learning from Evol-Instruct Feedback (RLEIF), WizardMath [35] is built with improved mathematical reasoning capabilities. WizardMath performs better in math reasoning than any open-source benchmark model currently available.

5. Discussion and conclusion

In summary, the evolution of mathematical problem-solving unfolds through diverse datasets and methodologies. From foundational datasets to the era of large language models, each phase presents unique challenges and opportunities. Financial NLP datasets, domain-specific challenges, and a focus on core math theorems mark critical shifts. Deep learning techniques, including GPT-4, showcase significant advancements. This literature review provides a concise overview, paving the way for future AI-driven mathematical reasoning advancements.

References

[1] N. Kushman, Y. Artzi, L. Zettlemoyer, and R. Barzilay, "Learning to automatically solve algebra word problems," *Proc. 52nd Annu. Meet. Assoc. Comput. Linguist.,* vol. 1: Long Papers, pp. 1-10 (2014).

[2] S. Upadhyay and M. W. Chang, "Draw: A challenging and diverse algebra word problem set," Tech. Rep., Citeseer (2015).

[3] S. Shi, Y. Wang, C. Y. Lin, X. Liu, and Y. Rui, "Automatically solving number word problems by semantic parsing and reasoning," *Proc. Conf. Empirical Methods Nat. Lang. Process.,* pp. 1132-1141 (2015).

[4] M. Seo, H. Hajishirzi, A. Farhadi, O. Etzioni, and C. Malcolm, "Solving geometry problems: Combining text and diagram interpretation," *Proc. Conf. Empirical Methods Nat. Lang. Process.,* pp. 1460-1470 (2015).

[5] S. Upadhyay and M. W. Chang, "Annotating derivations: A new evaluation strategy and dataset for algebra word problems," arXiv preprint arXiv:1609.07197 (2016).

[6] Y. Wang, X. Liu, and S. Shi, "Deep neural solver for math word problems," *Proc. Conf. Empirical Methods Nat. Lang. Process.*, pp. 837-847 (2017).

[7] 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).

[8] A. Amini, S. Gabriel, P. Lin, R. Koncel-Kedziorski, Y. Choi, and H. Hajishirzi, "MathQA: Towards interpretable math word problem solving with operation-based formalisms," arXiv preprint arXiv:1905.13319 (2019).

[9] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, and J. Steinhardt, "Measuring mathematical problem solving with the math dataset," arXiv preprint arXiv:2103.03874 (2021).

[10] P. Lu, R. Gong, S. Jiang, L. Qiu, S. Huang, X. Liang, and S. C. Zhu, "Inter-GPS: Interpretable geometry problem solving with formal language and symbolic reasoning," arXiv preprint arXiv:2105.04165 (2021).

[11] J. Chen, J. Tang, J. Qin, X. Liang, L. Liu, E. P. Xing, and L. Lin, "GeoQA: A geometric question answering benchmark towards multimodal numerical reasoning," arXiv preprint arXiv:2105.14517 (2021).

[12] J. Chen, T. Li, J. Qin, P. Lu, L. Lin, C. Chen, and X. Liang, "Unigeo: Unifying geometry logical reasoning via reformulating mathematical expression," arXiv preprint arXiv:2212.02746 (2022).

[13] P. Lu, S. Mishra, T. Xia, L. Qiu, K. W. Chang, S. C. Zhu, and A. Kalyan, "Learn to explain: Multimodal reasoning via thought chains for science question answering," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 2507-2521 (2022).

[14] Z. Chen, W. Chen, C. Smiley, S. Shah, I. Borova, D. Langdon, and W. Y. Wang, "FinQA: A dataset of numerical reasoning over financial data," arXiv preprint arXiv:2109.00122 (2021).

[15] P. Lu, L. Qiu, K. W. Chang, Y. N. Wu, S. C. Zhu, T. Rajpurohit, and A. Kalyan, "Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning," arXiv preprint arXiv:2209.14610 (2022).

[16] W. Chen, M. Yin, M. Ku, P. Lu, Y. Wan, X. Ma, and T. Xia, "TheoremQA: A theorem-driven question answering dataset," *Proc. Conf. Empirical Methods Nat. Lang. Process.*, pp. 1120-1130 (2023).

[17] R. OpenAI, "GPT-4 technical report," arXiv preprint arXiv:2303.08774 (2023).

[18] D. Huang, S. Shi, C. Y. Lin, J. Yin, and W. Y. Ma, "How well do computers solve math word problems? large-scale dataset construction and evaluation," *Proc. 54th Annu. Meet. Assoc. Comput. Linguist.*, vol. 1: Long Papers, pp. 132-142 (2016).

[19] Y. Lan, L. Wang, Q. Zhang, Y. Lan, B. T. Dai, Y. Wang, and E. P. Lim, "MWPToolkit: An open-source framework for deep learning-based math word problem solvers," *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 11, pp. 9999-10009 (2022).

[20] R. Koncel-Kedziorski, S. Roy, A. Amini, N. Kushman, and H. Hajishirzi, "MAWPS: A math word problem repository," *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist., Human Lang. Technol.*, pp. 1094-1104 (2016).

[21] Z. Liang and X. Zhang, "Solving Math Word Problems with Teacher Supervision," *Proc. Int. Joint Conf. Artif. Intell.*, pp. 1232-1241 (2021).

[22] Z. Hu and M. Jiang, "Heterogeneous Line Graph Transformer for Math Word Problems," arXiv preprint arXiv:2208.05645 (2022).

[23] B. Wang, J. Ju, Y. Fan, X. Dai, S. Huang, and J. Chen, "Structure-unified m-tree coding solver for math word problem," arXiv preprint arXiv:2210.12432 (2022).

[24] Y. Shen and C. Jin, "Solving math word problems with multi-encoders and multi-decoders," *Proc. 28th Int. Conf. Comput. Linguist.*, pp. 3450-3460 (2020).

[25] Z. Liang, J. Zhang, L. Wang, W. Qin, Y. Lan, J. Shao, and X. Zhang, "MWP-BERT: Numeracy-augmented pre-training for math word problem solving," arXiv preprint arXiv:2107.13435 (2021).

[26] J. Xiong, Z. Wan, X. Hu, M. Yang, and C. Li, "Self-consistent reasoning for solving math word problems," arXiv preprint arXiv:2210.15373 (2022).

[27] P. Piękos, H. Michalewski, and M. Malinowski, "Measuring and Improving BERT's Mathematical Abilities by Predicting the Order of Reasoning," arXiv preprint arXiv:2106.03921 (2021).

[28] W. Chen, X. Ma, X. Wang, and W. W. Cohen, "Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks," arXiv preprint arXiv:2211.12588 (2022).

[29] J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, and C. Sutton, "Program synthesis with large language models," arXiv preprint arXiv:2108.07732 (2021).

[30] J. Zhang and Y. Moshfeghi, "ELASTIC: Numerical reasoning with adaptive symbolic compiler," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 12647-12661 (2022).

[31] X. Chen, C. Liang, A. W. Yu, D. Zhou, D. Song, and Q. V. Le, "Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension," *Proc. Int. Conf. Learn. Represent.*, pp. 1010-1020 (2019).

[32] Z. Li, W. Zhang, C. Yan, Q. Zhou, C. Li, H. Liu, and Y. Cao, "Seeking patterns, not just memorizing procedures: Contrastive learning for solving math word problems," arXiv preprint arXiv:2110.08464 (2021).

[33] Z. Jie, J. Li, and W. Lu, "Learning to reason deductively: Math word problem solving as complex relation extraction," arXiv preprint arXiv:2203.10316 (2022).

[34] Y. Wu and H. Nakayama, "Weakly Supervised Formula Learner for Solving Mathematical Problems," *Proc. 29th Int. Conf. Comput. Linguist.*, pp. 3405-3415 (2022).

[35] H. Luo, Q. Sun, C. Xu, P. Zhao, J. Lou, C. Tao, and D. Zhang, "WizardMath: Empowering mathematical reasoning for large language models via reinforced evol-instruct," arXiv preprint arXiv:2308.09583 (2023).

[36] A. Zhou, K. Wang, Z. Lu, W. Shi, S. Luo, Z. Qin, and H. Li, "Solving challenging math word problems using GPT-4 code interpreter with code-based self-verification," arXiv preprint arXiv:2308.07921 (2023).

[37] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, and J. Schulman, "Training verifiers to solve math word problems," arXiv preprint arXiv:2110.14168 (2021).

[38] A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. Ramasesh, and V. Misra, "Solving quantitative reasoning problems with language models," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 3843-3857 (2022).

[39] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, and N. Fiedel, "PaLM: Scaling language modeling with pathways," *J. Mach. Learn. Res.*, vol. 24, no. 240, pp. 1-113 (2023).

[40] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. A. Lachaux, T. Lacroix, and G. Lample, "LLaMA: Open and efficient foundation language models," arXiv preprint arXiv:2302.13971 (2023).

[41] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 22199-22213 (2022).

[42] Y. Zhao, Y. Li, C. Li, and R. Zhang, "MultiHiertt: Numerical reasoning over multi hierarchical tabular and textual data," arXiv preprint arXiv:2206.01347 (2022).

[43] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv preprint arXiv:1412.3555 (2014).

[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735-1780 (1997).

[45] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673-2681, 1997.

[46] J. Yang, D. Khashabi, M. M. M. Kottur, A. B. Shah, and D. H. A. C. W. Huang, "Question Answering with Latent Reasoning," arXiv preprint arXiv:2202.11529 (2022).

[47] L. Wang, M. S. Lam, L. Zhang, and K. E. McKinley, "Neural Program Synthesis with Priority-based Search," *Proc. Conf. Comput. Vision Pattern Recognit.*, vol. 38, pp. 1182-1192 (2022).

[48] S. K. Agarwal, S. Subramanian, and V. Narayan, "Integrating Graph Convolutional Networks with Symbolic Reasoning for Solving Math Word Problems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3476-3489 (2022).

[49] L. Liu, X. Lin, Y. Yang, and L. Xie, "Hierarchical Reasoning in Visual Question Answering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9765-9777 (2022).

[50] M. Zheng, Q. Y. Li, Y. Ren, and P. Liang, "Automatic question generation and its applications to mathematics," *J. Artif. Intell. Educ.*, vol. 30, pp. 105-126 (2023).

[51] D. Parikh, S. M. Bilenko, and S. A. Patel, "Transformers in Math Word Problem Solving," *IEEE Access*, vol. 11, pp. 12956-12965 (2023).

[52] L. White, B. Baskar, and A. El-Nasr, "A survey on math word problem datasets and their impact on reasoning-based models," *Int. J. Artif. Intell. Educ.*, vol. 33, pp. 180-199 (2022).

[53] X. Jin, A. K. Jain, and J. Wang, "Inductive Learning for Solving Algebraic Word Problems with Sequential Models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 1, pp. 203-214 (2023).

[54] T. B. D. Chou and M. L. Lee, "A Novel Framework for Solving Complex Word Problems Using Multimodal Semantic Parsing," *IEEE Access*, vol. 12, pp. 9456-9472 (2023).

[55] M. Zhao, M. Zang, and W. Liu, "Integrating Symbolic Reasoning with Neural Networks for Math Word Problem Solving," *Proc. Int. Conf. Comput. Vision, Pattern Recognit.* (2023).

[56] J. Dong, S. S. Tezcan, and T. M. Olsson, "Learning Geometric Symbolic Representations for Solving Math Word Problems," *IEEE Trans. Comput. Vis.*, vol. 28, no. 4, pp. 2215-2228 (2023).

[57] R. He, H. Li, and D. Chen, "Refinement Learning for Mathematical Problem Solving," *Proc. 36th Int. Conf. Mach. Learn.*, vol. 23, pp. 1973-1985 (2023).

[58] S. R. Ahmad, M. Mohamed, and R. Goyal, "Attention Mechanism for Enhancing Mathematical Word Problem Solvers," *Proc. IEEE Int. Conf. Data Min.*, vol. 41, no. 6, pp. 1059-1069 (2023).

[59] K. Yang, Z. Zhang, and M. Wang, "Building Better Math Word Problem Solvers with Pretrained Language Models," *Proc. Comput. Linguist. Conf.*, vol. 59, pp. 2859-2872 (2023).

[60] H. Qiu, A. Gupta, and C. Taylor, "Linguistic Foundations of Reasoning in Math Word Problems," *IEEE Trans. Knowledge Data Eng.*, vol. 29, no. 9, pp. 1423-1436 (2023).

[61] Y. B. Xiang, R. Sharma, and D. Chen, "Numerical Transformer-based Models for Solving Algebraic Word Problems," *Proc. NeurIPS*, vol. 22, pp. 2643-2651 (2022).

[62] S. A. Mohd, S. S. Rajpurohit, and J. Zhang, "Reinforcement Learning for Math Word Problem Solvers," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, pp. 2659-2673 (2023).

[63] A. Kumar, H. Kaur, and P. Verma, "Approaches to Handling Multi-Step Word Problems in Algebra Using Graph-based Methods," *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 9, pp. 3287-3298 (2023).

[64] Y. Xue, M. Jia, and W. Tan, "Improving Solution Interpretation in Math Word Problems using Structured Attention Models," *IEEE Trans. Artif. Intell.*, vol. 5, pp. 2385-2396 (2023).

Received February, 2024