A Survey of Deep Learning for Mathematical Reasoning

Pan Lu[1], **Liang Qiu**[1], **Wenhao Yu**[2], **Sean Welleck**[3*], **Kai-Wei Chang**[1*]
[1]UCLA, [2]University of Notre Dame, [3]University of Washington
https://github.com/lupantech/dl4math

Abstract

Mathematical reasoning is a fundamental aspect of human intelligence and is applicable in various fields, including science, engineering, finance, and everyday life. The development of artificial intelligence (AI) systems capable of solving math problems and proving theorems in language has garnered significant interest in the fields of machine learning and natural language processing. For example, mathematics serves as a testbed for aspects of reasoning that are challenging for powerful deep learning models, driving new algorithmic and modeling advances. On the other hand, recent advances in large-scale neural language models have opened up new benchmarks and opportunities to use deep learning for mathematical reasoning. In this survey paper, we review the key tasks, datasets, and methods at the intersection of mathematical reasoning and deep learning over the past decade. We also evaluate existing benchmarks and methods, and discuss future research directions in this domain.

1 Introduction

"*The study of mathematics, like the Nile, begins in minuteness but ends in magnificence.*"
— Charles Caleb Colton, English writer

Mathematical reasoning is a key aspect of human intelligence that enables us to comprehend and make decisions based on numerical data and language. It is applicable in various fields, including science, engineering, finance, and everyday life, and encompasses a range of abilities, from basic skills such as pattern recognition and numerical operations to more advanced skills like problem-solving, logical reasoning, and abstract thinking. The development of artificial intelligence (AI) systems capable of solving math problems and proving theorems in language has been a long-standing focus of research in the fields of machine learning and natural language processing (NLP), dating back to the 1960s (Feigenbaum et al., 1963; Bobrow, 1964). In recent years, there has been a surge of interest in this area: for instance, the number of papers has grown from approximately 10 in 2018 to 66 in 2022 (see Figure 3 in the Appendix).

As deep learning continues to revolutionize NLP tasks such as question answering and machine translation (Sutskever et al., 2014; Devlin et al., 2019), it has also made significant strides in the field of mathematical reasoning (Wang et al., 2017; Yang and Deng, 2019; Geva et al., 2020; Wei et al., 2022). However, despite the impressive capabilities of these models, there is still a lack of a clear taxonomy of the different types of mathematical reasoning tasks and the specific capabilities required of deep learning models to solve them.

Previous literature has been limited to the discussion of specific aspects, such as solving math word problems (Bhattacharya, 2017; Zhang et al., 2019; Ughade and Kumbhar, 2019), representing numbers representation (Thawani et al., 2021), or solving informal problems (Meadows and Freitas, 2022). Additionally, with the recent advancements in large language models like GPT-3 (Brown et al., 2020), there is a growing need to understand the capabilities and limitations of these models in the context of mathematical reasoning. This is where a comprehensive survey of this rapidly advancing domain becomes crucial, as it can provide an overview of the current state and limitations of the field, and indicate further research areas.

In this paper, we survey over 180 papers from the NLP and AI communities in the field of deep learning for mathematical reasoning. We study various types of mathematical reasoning problems, such as math word problems, theorem proving, geometry problem solving, math question answering, and other quantitative problems (§2, §A). Additionally, we explore different deep learning architectures for mathematical reasoning, including neural networks

[*] denotes co-senior authors.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 14605–14631
July 9-14, 2023 ©2023 Association for Computational Linguistics

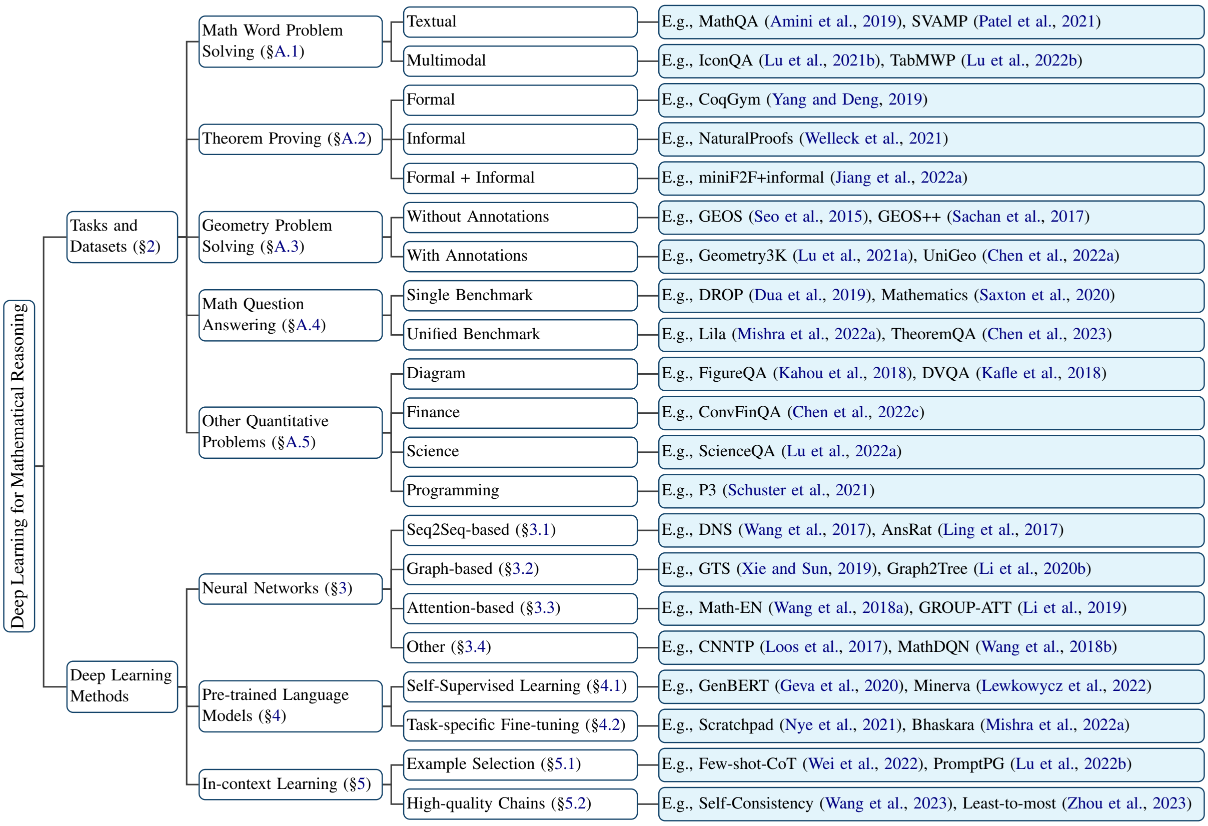

Deep Learning for Mathematical Reasoning
Tasks and Datasets (§2)
Math Word Problem Solving (§A.1)
Textual
E.g., MathQA (Amini et al., 2019), SVAMP (Patel et al., 2021)
Multimodal
E.g., IconQA (Lu et al., 2021b), TabMWP (Lu et al., 2022b)
Theorem Proving (§A.2)
Formal
E.g., CoqGym (Yang and Deng, 2019)
Informal
E.g., NaturalProofs (Welleck et al., 2021)
Formal + Informal
E.g., miniF2F+informal (Jiang et al., 2022a)
Geometry Problem Solving (§A.3)
Without Annotations
E.g., GEOS (Seo et al., 2015), GEOS++ (Sachan et al., 2017)
With Annotations
E.g., Geometry3K (Lu et al., 2021a), UniGeo (Chen et al., 2022a)
Math Question Answering (§A.4)
Single Benchmark
E.g., DROP (Dua et al., 2019), Mathematics (Saxton et al., 2020)
Unified Benchmark
E.g., Lila (Mishra et al., 2022a), TheoremQA (Chen et al., 2023)
Other Quantitative Problems (§A.5)
Diagram
E.g., FigureQA (Kahou et al., 2018), DVQA (Kafle et al., 2018)
Finance
E.g., ConvFinQA (Chen et al., 2022c)
Science
E.g., ScienceQA (Lu et al., 2022a)
Programming
E.g., P3 (Schuster et al., 2021)
Deep Learning Methods
Neural Networks (§3)
Seq2Seq-based (§3.1)
E.g., DNS (Wang et al., 2017), AnsRat (Ling et al., 2017)
Graph-based (§3.2)
E.g., GTS (Xie and Sun, 2019), Graph2Tree (Li et al., 2020b)
Attention-based (§3.3)
E.g., Math-EN (Wang et al., 2018a), GROUP-ATT (Li et al., 2019)
Other (§3.4)
E.g., CNNTP (Loos et al., 2017), MathDQN (Wang et al., 2018b)
Pre-trained Language Models (§4)
Self-Supervised Learning (§4.1)
E.g., GenBERT (Geva et al., 2020), Minerva (Lewkowycz et al., 2022)
Task-specific Fine-tuning (§4.2)
E.g., Scratchpad (Nye et al., 2021), Bhaskara (Mishra et al., 2022a)
In-context Learning (§5)
Example Selection (§5.1)
E.g., Few-shot-CoT (Wei et al., 2022), PromptPG (Lu et al., 2022b)
High-quality Chains (§5.2)
E.g., Self-Consistency (Wang et al., 2023), Least-to-most (Zhou et al., 2023)

Figure 1: Taxonomy of deep learning for mathematical reasoning. The associated tasks are elaborated in §2, with a comprehensive dataset list found in §A. Deep learning methods are further discussed in §3, §4, and §5.

(§3), pre-trained language models (§4), and recent in-context learning for large language models (§5).

We also analyze existing benchmarks and find that there is less focus on multi-modal and low-resource settings (§6.1). Our evidence-based studies suggest that current numeracy representations are insufficient and deep learning methods are inconsistent for mathematical reasoning (§6.2). Following this, we suggest future research directions related to generalization and robustness, trustworthy reasoning, learning from feedback, and multimodal mathematical reasoning (§7).

2 Mathematical Reasoning Tasks

In this section, we briefly introduce different tasks for mathematical reasoning. A detailed summary and discussion of commonly used datasets can be found in Table 7 and Appendix A.

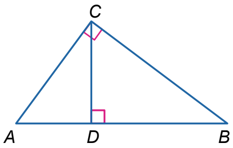
Math Word Problem Solving. Developing algorithms to automatically solve math word problems (MWPs) has been of interest to NLP researchers for decades (Feigenbaum et al., 1963; Bobrow, 1964). An example of a MWP is shown in Table 1. A question involves four basic arithmetic operations with single or multiple operation steps. The challenge posed by MWPs lies in the need for language comprehension, semantic parsing, and the application of multiple mathematical reasoning skills.

Question: Bod has 2 apples and David has 5 apples. How many apples do they have in total?
Rationale: $x = 2 + 5$
Solution: 7

Table 1: A typical math word problem.

Theorem Proving. Automating theorem proving is a long-standing challenge in AI (Newell et al., 1957; Feigenbaum et al., 1963). The problem is to demonstrate the truth of a mathematical claim (a *theorem*) through a sequence of logical arguments (a *proof*). Theorem proving tests various skills, such as choosing effective multi-step strategies, using background knowledge, and performing symbolic manipulations.

Geometry Problem Solving. Automated geometry problem solving (GPS) is also a long-standing mathematical reasoning task (Gelernter et al., 1960; Wen-Tsun, 1986). As shown in Figure 2, a geometry problem consists of a textual description and a diagram. The multimodal inputs describe the entities, attributes, and relationships of geometric elements, and the goal is to find the numeric solution to an unknown variable.



Question: In triangle ABC, $AD = 3$ and $BD = 14$. Find CD.

Choices: (A) 6.0 (B) 6.5 (C) 7.0 (D) 8.5

Answer: (B) 6.5

Figure 2: An example of geometry problems.

Math Question Answering. There is a wide range of question answering (QA) benchmarks that center around mathematical reasoning, which we refer to as math question answering (MathQA). For example, DROP (Dua et al., 2019) is a MathQA dataset that requires discrete reasoning to answer questions such as "Which kicker kicked the most field goals?" over the content of paragraphs.

3 Neural Networks for Mathematical Reasoning

Neural networks have become a popular tool in the field of mathematical reasoning, mirroring their success in NLP. In recent years, a number of different neural network architectures have been proposed for mathematical reasoning tasks, including Seq2Seq-based networks, graph-based networks, and attention-based networks. These methods are outlined in more detail in Table 8 in the Appendix.

3.1 Seq2Seq-based Networks for Math

Sequence-to-sequence (Seq2Seq) (Sutskever et al., 2014) neural networks have been successfully applied to mathematical reasoning tasks, such as math word problem solving (Wang et al., 2017), theorem proving (Yang and Deng, 2019), geometry problem solving (Robaidek et al., 2018), and math question answering (Tafjord et al., 2019). A Seq2Seq model uses an encoder-decoder architecture and usually formalizes mathematical reasoning as a sequence generation task. The basic idea behind this approach is to map an input sequence (e.g. a mathematical problem) to an output sequence (e.g. an equation, program, and proof). Common encoders and decoders include Long Short Term Memory network (LSTM) (Hochreiter and Schmidhuber, 1997), Gated Recurrent Unit (GRU) (Cho et al., 2014), and their bidirectional variants: BiLSTM and BiGRU. A large amount of work has shown the performance advantage of Seq2Seq models over previous statistical learning approaches (Ling et al., 2017; Wang et al., 2018a; Huang et al., 2018; Wang et al., 2019; Li et al., 2019).

3.2 Graph-based Networks for Math

Seq2Seq approaches show their advantages of generating mathematical expressions without relying on hand-crafted features. It is noteworthy that mathematical expressions can be represented as tree-based structures, such as abstract syntax trees (ASTs) and graph-based structures, which capture the structural information in the expressions. However, Seq2Seq methods do not explicitly this important information. To address this limitation, graph-based neural networks have been developed to explicitly model the structure within expressions. Sequence-to-tree (Seq2Tree) models explicitly model the tree structure when encoding the output sequences (Xie and Sun, 2019; Wu et al., 2020; Zaporojets et al., 2021; Qin et al., 2021). For example, Liu et al. (2019a) devise a Seq2Tree model to better use information from an equation's AST. Seq2DAG (Cao et al., 2021), instead, applies a sequence-to-graph (Seq2Graph) framework when generating the equations since the graph decoder is able to extract complex relationships among multiple variables. The graph-based information can also be embedded when encoding the input mathematical sequences (Zhang et al., 2020b; Shen and Jin, 2020; Li et al., 2020b; Wu et al., 2021a).

3.3 Attention-based Networks for Math

The attention mechanism has been successfully applied to NLP (Bahdanau et al., 2015) and vision problems (Xu et al., 2015; Woo et al., 2018), taking into account the hidden vectors of the inputs during the decoding processing. Recently, researchers have been exploring its usefulness in mathematical reasoning tasks, as it can be used to identify the most important relationships between mathematical concepts. For instance, MATH-EN (Wang et al., 2018a) is a math word problem solver which benefits from long-distance dependency information learned by self-attention. Attention-based methods have also been applied to other mathematical reasoning tasks such as geometry problems solving (Robaidek et al., 2018; Chen et al., 2021a) and theorem proving (Yang and Deng, 2019). Various attention mechanisms have been studied to extract better representations, such as Group-ATT (Li et al., 2019) which uses different multi-head attention to extract various types of MWP features, and graph attention which is applied to extract knowledge-aware information in (Wu et al., 2020).

3.4 Other Neural Networks for Math

Deep learning approaches to mathematical reasoning tasks can also make use of other neural networks, such as convolutional neural networks (CNN) and multimodal networks. Some work encodes the input text using a convolutional neural network architecture, giving the model the ability to capture long-term relationships between symbols in the input (Gehring et al., 2017; Wang et al., 2018a,a; Robaidek et al., 2018; Alemi et al., 2016; Loos et al., 2017). For example, the first application of deep neural networks for theorem proving is proposed in (Alemi et al., 2016), which relies on convolutional networks for premise selection.

Multimodal mathematical reasoning tasks, such as geometry problem solving and diagram-based mathematical reasoning, are formalized as visual question answer (VQA) problems (Kafle et al., 2018; Chen et al., 2021a; Lu et al., 2021b). In this domain, visual inputs are encoded using ResNet (He et al., 2016) or Faster-RCNN (Ren et al., 2015), while textual representations are obtained via GRU or LTSM. Subsequently, the joint representation is learned using multimodal fusion models, such as BAN (Kim et al., 2018), FiLM (Perez et al., 2018), and DAFA (Gao et al., 2019).

Other deep neural network structures can also be used in mathematical reasoning. A Graph Neural Network (GNN) is employed for geometry problem parsing in Zhang et al. (2022), taking advantage of its success in spatial reasoning. WaveNet has been applied to theorem proving (Loos et al., 2017; Bansal et al., 2019), due to its ability to address longitudinal time-series data. Furthermore, Transformers are found to outperform GRU in generating mathematical equations in DDT (Meng and Rumshisky, 2019). Finally, MathDQN (Wang et al., 2018b) is the first work to explore reinforcement learning for math word problem solving, taking advantage of its strong search capabilities.

4 Pre-trained Language Models for Mathematical Reasoning

Pre-trained language models (Devlin et al., 2019; Radford et al., 2020; Brown et al., 2020) have demonstrated remarkable performance gains on a wide range of NLP tasks. By pre-training on a large corpus of text, the models learn valuable world knowledge (Guu et al., 2020), which could be applied to downstream tasks. Similar ideas can be applied to math-related problems, and previous work has shown the promising performance of pre-trained language models in answering math word problems (Kim et al., 2020), assisting with theorem proving (Wu et al., 2022b), as well as solving other mathematical tasks (Charton, 2022).

However, though large language models excel in modeling natural language, there are several challenges to using them for mathematical reasoning. First, pre-trained language models are not specifically trained on mathematical data. This likely contributes to them being less proficient in math-related tasks compared to natural language tasks. There is also less mathematical or scientific data available for large-scale pre-training compared to text data. Second, the size of pre-trained models continues to grow, making it expensive to train the entire model from scratch for specific downstream tasks. Additionally, downstream tasks may deal with different input formats or modalities, such as structured tables (Zhao et al., 2022) or diagrams (Lu et al., 2021b). To address these challenges, researchers have to adjust pre-trained models by finetuning them on downstream tasks or adapting the neural architectures.

4.1 Self-Supervised Learning for Math

Self-supervised learning is a machine learning approach in which an algorithm learns to perform a task without being explicitly provided with labeled training data. Table 2 provides a list of language models pre-trained with self-supervised tasks for mathematical reasoning.

Model scale. There is a clear trend that pre-trained language models have become increasingly larger in the past few years (Devlin et al., 2019; Lewis et al., 2020; Raffel et al., 2020; Radford et al., 2020; Brown et al., 2020). A recent study (Liang et al., 2022a) shows that model scale within a model family reliably predicts model accuracy. The study also mentions an interesting thresholding effect: "all models that win head-to-head model comparisons for accuracy at a rate well above chance are at least 50B parameters". A similar size-growing trend can be observed in the field of mathematical reasoning with pre-trained language models. For example, MWP-BERT (Liang et al., 2022b) uses a backbone of BERT (110M) (Devlin et al., 2019) and RoBERTa (123M) (Liu et al., 2019b) for Math Word Problems. Most recently, Minerva (Lewkowycz et al., 2022), which is based on the PaLM (Chowdhery et al., 2022) pre-trained

Paper	Backbone	Size	Corpus	Pre-training task
GPT-*f* (Polu and Sutskever, 2020)	Transformer (2017)	774M	Math	Causal language modeling
LISA (Jiang et al., 2021)	Transformer (2017)	163M	Math	Causal language modeling
MATH-PLM (Hendrycks et al., 2021b)	GPT-2 (2020)	1.5B	Math	Causal language modeling
MWP-BERT (Liang et al., 2022b)	RoBERTa (2019b)	123M	Math	8 numeracy augmented tasks
TaPEx (Liu et al., 2022b)	BART (2020)	406M	SQL	Query result generation
HTPS (Lample et al., 2022)	Transformer (2017)	600M	Math	Masked Seq2Seq modeling
Thor (Jiang et al., 2022b)	Transformer (2017)	700M	Github, arXiv	Causal language modeling
PACT (Han et al., 2022)	Transformer (2017)	837M	Math	Masked/Causal language modeling
Minerva (Lewkowycz et al., 2022)	PaLM (2022)	540B	Science & Math	Causal language modeling
GenBERT (Geva et al., 2020)	BERT (2019)	110M	Number, Text	Masked/Causal language modeling
NF-NSM (Feng et al., 2021)	RoBERTa (2019b)	110M	Number	Number prediction
LIME (Wu et al., 2021d)	Transformer (2017)	11B	Math	Causal language modeling
Set (Wu et al., 2022c)	T5 (2020)	60M	Math	Unique token generation

Table 2: Comparison of pre-training language models for mathematical reasoning.

Paper	Backbone	Task
EPT (2020)	ALBERT (2019)	MWP
Generate & Rank (2021)	BART (2020)	MWP
RPKHS (2021b)	RoBERTa (2019b)	MWP
PatchTRM (2021b)	ResNet+BERT (2019)	MWP
GSM8K-PLM (2021)	GPT-3 (2020)	MWP
BERT-TD+CL (2022b)	BERT (2019)	MWP
DeductReasoner (2022)	RoBERTa (2019b)	MWP
Self-Sampling (2023)	GPT-Neo (2020)	MWP
Bhaskara (2022a)	GPT-Neo (2020)	MWP
miniF2F-PLM (2022)	GPT-*f* (2020)	TP
NaturalProver (2022a)	GPT-3 (2020)	TP
Inter-GPS (2021a)	BART (2020)	GPS
UniGeo (2022a)	VL-T5 (2021)	GPS
DPE-NGS (2022)	RoBERTa (2019b)	GPS
Aristo (2020)	RoBERTa (2019b)	MathQA
FinQANet (2021c)	RoBERTa (2019b)	MathQA
TAGOP (2021)	RoBERTa (2019b)	MathQA
MT2Net (2022)	RoBERTa (2019b)	MathQA
Scratchpad (2021)	Transformer (2017)	Mixed
LAMT (2022)	Transformer (2017)	Mixed

Table 3: Finetuned pre-trained language models for downstream mathematical reasoning tasks.

language model, has a size up to 540B parameters.

Pre-training corpus. There are generally two types of pre-training corpus for mathematical language models. (i) Curated datasets from openly accessible sources. For example, Hendrycks et al. (2021b) present the first large-scale mathematics pre-training dataset with step-by-step solutions in natural language and LaTeX, called the Auxiliary Mathematics Problems and Solutions (AMPS). AMPS consists of Khan Academy and Mathematica data. Minerva (Lewkowycz et al., 2022) collects a high-quality dataset containing scientific and mathematical data, which contains 38.5B tokens from webpages filtered for mathematical content and from papers submitted to the arXiv preprint server. Thor (Jiang et al., 2022b) pre-trains a language model on the GitHub + arXiv subsets of The Pile (Gao et al., 2020). (ii) Synthetic datasets based on templates or interaction with engines. Recent work (Wu et al., 2021d; Krishna et al., 2021; Ri and Tsuruoka, 2022; Anderson and Farrell, 2022; Wu et al., 2022c) shows that pre-training on data that is fully synthetically generated—synthetic pre-training can actually provide substantial gains. Representative work includes TaPEX (Liu et al., 2022b), which obtains a pre-training corpus by automatically synthesizing executable SQL queries and their execution outputs. LISA (Jiang et al., 2021) extracts lemmas and theorems by interacting with the Isabelle standard library and the Archive of Formal Proofs. GenBERT (Geva et al., 2020) generates numerical and textual pre-training datasets based on manually crafted and extracted templates.

Pre-training tasks. General pre-training language models have two typical self-supervised learning tasks: (i) Masked Language Modeling (MLM), where it randomly masks a portion of words in each sequence to predict the outcome; (ii) Causal Language Modeling (CLM), where the model is trained to predict the next token in a sequence of tokens. Following the same paradigm, researchers pre-train language models with MLM and CLM tasks on mathematical or scientific corpora for downstream tasks (Polu and Sutskever, 2020; Hendrycks et al., 2021b; Han et al., 2022; Jiang et al., 2022b).

There is also recent work that designs customized tasks to inject mathematical reasoning capabilities into language models. For instance, Liang et al. (2022b) pre-train language models with a suite of 8 numeracy-augmented tasks with consideration of reasoning logic and numerical properties. LIME (Wu et al., 2021d) proposes synthetic pre-training tasks to learn three reasoning primitives: deduction, induction, and abduction before learning more complex reasoning skills, which also be regarded as a form of curriculum learning.

4.2 Task-specific Fine-tuning for Math

Task-specific fine-tuning is a technique to improve the performance of a pre-trained language model on a specific task. This is also a common practice when there is not enough data for training the large models from scratch. As shown in Table 3, existing work fine-tunes pre-trained language models on a variety of downstream tasks, such as math word problems (Kim et al., 2020; Shen et al., 2021), MathQA (Zhao et al., 2022), geometry problem solving (Lu et al., 2021a), linear algebra (Charton, 2022), and theorem proving (Welleck et al., 2022a). Apart from fine-tuning the model parameters, some work also uses pre-trained language models as encoders and ensembles them with other modules for downstream tasks (Lu et al., 2021b).

5 In-context Learning for Mathematical Reasoning

Large language models (LLMs), such as GPT-3 (Brown et al., 2020), have recently revolutionized the field of natural language processing (NLP), especially on account of their powerful few-shot in-context learning capabilities (Brown et al., 2020). In-context Learning (ICL) enables LLMs to perform target tasks by providing some task examples as conditions at inference time, without updating model parameters (Radford et al., 2020; Brown et al., 2020). ICL allows users to quickly build models for new use cases without worrying about fine-tuning and storing a large amount of new parameters for each task, so it is widely used in few-shot settings nowadays (Min et al., 2022).

An in-context example typically contains an input-output pair with some prompt words, e.g., *Please select the largest number from the list. Input: [2, 4, 1, 5, 8]. Output: 8*, and few-shot works by giving multiple examples, and then a final input example, where the model is expected to predict the output. However, such standard few-shot promptings, in which the LLM is given in-context examples of input–output pairs in front of test-time examples, have not yet proved sufficient to achieve high performance on challenging tasks such as mathematical reasoning (Rae et al., 2021).

Chain-of-thought prompting (CoT) (Wei et al., 2022) leverages intermediate natural language rationales as prompts to enable LLMs to first generate *reasoning chains* and then predict an answer for an input question. For example, a CoT prompt for solving the math word problem could be

> **Question:** Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. Then, how many tennis balls does Roger have now?
> **Answer:** *Roger started with 5 balls. 2 cans of 3 tennis balls each are 6 tennis balls. 5 + 6 = 11.* The answer is <u>11</u>.

Apart from Kojima et al. (2022) showing that LLMs are decent zero-shot reasoners when given the "Let's think step by step!" prompt, most of the recent work has focused on how to improve chain-of-thought reasoning under the few-shot setting. This work is mainly divided into two parts, (i) selecting better in-context examples and (ii) creating better reasoning chains.

5.1 In-context Example Selection

Early chain-of-thought work randomly or heuristically selects in-context examples. However, recent studies have shown that this type of few-shot learning can be highly unstable across different selections of in-context examples (Rubin et al., 2022; Liu et al., 2022a). Therefore, which in-context reasoning examples make the most effective prompts is still an unknown problem in the literature. To address the limitation, recent work has investigated various methods to optimize the in-context examples selection process (Rubin et al., 2022; Zhang et al., 2023; Lu et al., 2022b; Yu et al., 2023; Fu et al., 2023). For example, Rubin et al. (2022) attempt to address this issue by retrieving semantically similar examples. In addition, Fu et al. (2023) propose complexity-based prompting, which chooses examples with complex reasoning chains, i.e., chains with more reasoning steps, as the prompt. PromptPG (Lu et al., 2022b) learns to select optimal in-context examples via reinforcement learning (RL) from a candidate pool.

5.2 High-quality Reasoning Chains

Early chain of thought work (e.g., Wei et al. (2022)) mainly relies on a single human-annotated reasoning chain as a prompt. However, manually creating reasoning chains has two disadvantages. First, as tasks become more complex, current models may not be sufficient to learn to perform all necessary reasoning steps and cannot easily generalize to different tasks. Second, a single decoding process is vulnerable to incorrect inference steps, leading to an incorrect prediction as the final answer. To address this limitation, recent studies mainly fo-

Models	Engine (best performed)	ICL source	Rationale type	Rationale source	Post method
Few-shot-CoT (Wei et al., 2022)	PaLM (540B)	Random	Language	Hand-crafted	-
Self-Consistency-CoT (Wang et al., 2023)	Codex (175B)	Random	Language	Hand-crafted	Self-consistency
Least-to-most CoT (Zhou et al., 2023)	Codex (175B)	Random	Language	Hand-crafted	-
PromptPG-CoT (Lu et al., 2022b)	GPT-3 (175B)	RL	Language	Hand-crafted	-
Retrieval-CoT (Zhang et al., 2023)	GPT-3 (175B)	Retrival	Language	Auto-generated	-
Auto-CoT (Zhang et al., 2023)	Codex (175B)	Clustering	Language	Auto-generated	-
Complexity-CoT (Fu et al., 2023)	GPT-3 (175B)	Complexity	Language	Hand-crafted	Self-consistency
Few-shot-PoT (Chen et al., 2022b)	GPT-3 (175B)	Random	Code	Hand-crafted	-

Table 4: In-context learning with large language models for mathematical reasoning. For GPT-3, all papers use the text-davinci-002 version; for Codex, all papers use the code-davinci-002. RL is short for reinforcement learning.

cus on two aspects, (i) hand-crafting more complex demonstrations, which we refer to as *process-based approaches* (Zhou et al., 2023; Chen et al., 2022b), (ii) leveraging ensemble-like methods, which we refer to as *outcome-based approaches* (Wang et al., 2023; Li et al., 2022a).

Process-based approaches aim to improve the chain-of-thought reasoning quality, especially for complex reasoning tasks. In least-to-most prompting (Zhou et al., 2023), the problem-solving process is implemented through two-stage prompting: (i) reducing a complex problem into a list of subproblems; (ii) solving these sub-problems sequentially, so that solving a given sub-problem is facilitated by the answers to previously solved subproblems. Similarly, Khot et al. (2022) leverage diverse decomposition structures and use different prompts to answer each sub-question. Apart from these multi-step reasoning methods, Chen et al. (2022b); Gao et al. (2022) propose program-of-thoughts (PoT), an alternative solution that uses large language models to express the reasoning process as a program. The computation is then relegated to an external computer, which executes the generated programs to derive the answer. A more recent work, Chameleon (Lu et al., 2023), integrates different tools to enhance the abilities of LLMs for compositional reasoning.

Outcome-based approaches acknowledge the potential incorrectness of an individual reasoning path, and instead use multiple reasoning paths (Wang et al., 2023; Li et al., 2022a). Self-consistency (Wang et al., 2023) generates a set of reasoning paths by sampling from the language model, and marginalizes out the reasoning paths by choosing the most common answer. In addition to using sampling with a single prompt to produce multiple reasoning paths, Li et al. (2022a) propose to introduce diverse prompts through "self-teaching", as a complementary solution to produce a higher degree of diversity.

6 Discussion and Findings

6.1 Analysis of Benchmarks

The multi-modal setting is underexplored but is gaining increasing attention. Most existing benchmarks for mathematical reasoning have targeted the textual-only modality. However, visual elements can provide a rich source of quantitative information, making multi-modal datasets beneficial for reasoning over quantitative relations in natural images (Lu et al., 2022a), abstract diagrams (Lu et al., 2021b), figures (Kahou et al., 2018), and charts (Kafle et al., 2018). Tables, which are commonly found in daily documents and contain hierarchically structured information, have also been the focus of tasks that require quantitative reasoning over textual and tabular context (Chen et al., 2021c; Zhu et al., 2021; Zhao et al., 2022; Lu et al., 2022b). In addition, recent datasets have been developed for mathematical reasoning grounded on conversations (Sun et al., 2019; Zhang et al., 2021; Chen et al., 2022c), as well as reports (Chen et al., 2022c).

Pioneering work is emerging in the exploration of low-resource settings. Despite the creation of various datasets, mathematical reasoning in low-resource settings remains largely under-explored. Pioneering research has developed mathematical reasoning benchmarks for financial (Chen et al., 2021c; Zhu et al., 2021; Zhao et al., 2022) and scientific domains (Lu et al., 2022a). Additionally, there have been attempts to build non-English datasets for Chinese (Wang et al., 2017; Qin et al., 2020; Yu et al., 2021a) and Arabic (Alghamdi et al., 2022) for mathematical reasoning.

Diverse rationale annotations have been widely explored. Complex reasoning usually involves multiple steps to arrive at the final answer. To bridge this gap, datasets annotated with intermediate rationales such as logic forms (Tafjord et al.,

	T5	UnifiedQA	GPT-3	GPT-3
	(Large)	(Large)	(davinci-002)	(davinci-003)
3 balls + 5 balls =	✗	5 balls	8 balls	8 balls
23 balls + 145 balls =	✗	✗	58 balls	168 balls
23 balls + 1,855 balls =	✗	✗	2,878 balls	2,988 balls

Table 5: Language models struggle with large numbers.

2019; Lu et al., 2021a), programs (Amini et al., 2019; Chen et al., 2021c,a; Cao and Xiao, 2022; Chen et al., 2022a), and reasoning graphs (Zhang et al., 2021) have been proposed to train models for complex reasoning tasks. Python programs are used as reasoning annotations in (Austin et al., 2021; Mishra et al., 2022a) due to their enhanced accessibility and readability. To imitate the reasoning process of a human, a more recent trend is to annotate solutions in natural language (Ling et al., 2017; Cobbe et al., 2021; Lu et al., 2022b; Hendrycks et al., 2021b; Lu et al., 2022a).

6.2 Analysis of Deep Learning Methods

Is the current representation of numeracy sufficient? The standard practice for deep learning techniques is to treat numbers in the same way as words. Early neural network methods create a vocabulary that maps input words and numbers to token IDs, resulting in less frequent numbers being collapsed into an "UNK" token. Recent language models use subword tokenization techniques (Wu et al., 2016; Sennrich et al., 2016) to split numbers into atomic tokens. Recent studies have shown that these tokenization approaches are suboptimal (Wallace et al., 2019; Lin et al., 2020; Zhang et al., 2020d; Thawani et al., 2022).

Two numbers on the same or close number line could have surface forms with no shared common tokens. For example, a number like 1598 is tokenized as "15" and "98" in GPT-3, while another format like $1,598$ is split as three different tokens: "1", ",", and "598". This lack of consistent representation can make it difficult for deep learning models to effectively process numbers, especially when compared to pure text. The insufficient representations of numbers can lead to out-of-distribution (OOD) problems. Table 5 provides examples of where language models tend to struggle with large numbers. Although increasing model scales could help, even the state-of-the-art large language model GPT-3 performs poorly when reasoning over large numbers. Some recent work suggests that using scientific notation (Zhang et al., 2020d) and digit-level decomposition (Geva et al., 2020) may be helpful in improving numeracy representation, but this remains an open problem.

Problems	**GPT-3** (text-davinci-002)
John had 8 balls and he gave 3 to Mary. How many balls does John have now?	John has 5 balls.
John had 3 apples. John had 8 balls and he gave 3 to Mary. How many balls does Mary have now?	Mary has 5 balls.
John had 8 balls and he gave 3 to Mary. Who has more balls now?	John has more balls.
John had 8 balls and he gave 3 to Mary. Does John have more balls now?	No, John has 5 balls now.
John had 8 balls and he gave 4 to Mary. Does John have more balls now?	No, John has 4 balls now.
John had 8 balls and he gave 4 to Mary. Who has more balls now?	John has more balls.

Table 6: Examples where large language models are not consistent for mathematical reasoning.

Are deep learning methods consistent for mathematical reasoning? Recent developments in deep learning have led to impressive results on various mathematical reasoning tasks. The zero-shot-CoT Minerva 540B achieves a score of 75.0% on the MMLU-STEM benchmark (Hendrycks et al., 2021a), which assesses multitask reasoning ability in the fields of science, technology, engineering, and mathematics (STEM) at both high school and college levels. Similarly, few-shot-CoT GPT-3 175B achieves a high accuracy of 93.0% on the MultiArith task. However, the question remains as to whether these methods are sufficiently advanced to tackle more complex problems.

There is strong evidence that deep learning methods for mathematical reasoning are not robust and susceptible to adversarial attacks (Lin et al., 2020; Patel et al., 2021; Mishra et al., 2022b,a; Welleck et al., 2022b). The SVAMP (Patel et al., 2021) dataset is a collection of one-unknown arithmetic word problems up to grade 4, with slight word variations from previous datasets. It is surprising that current state-of-the-art (SOTA) methods perform poorly on this dataset, with Graph2Tree achieving only a 43.8% accuracy and zero-shot-CoT GPT-3 (175B) only reaching 63.7%, which is just above an "F" grade. Table 6 also shows the inconsistent performance of the zero-shot GPT-3 model in scenarios with slightly different descriptions, while human performance remains unchanged. This indicates a lack of consistency in the mathematical reasoning ability of SOTA large language models.

7 Future Work

7.1 Generalization and Robustness

Despite impressive progress, neural models commonly display generalization and robustness failures on reasoning tasks. For example, above we discussed difficulties in generalizing to larger numbers (Table 5) or remaining robust to nearby problems (Table 6), while others identify failures in generalizing to longer problems than those observed in training (e.g., Anil et al. (2022)). One direction is to explore new inference-time (Jung et al., 2022; Mitchell et al., 2022) or fine-tuning (Anil et al., 2022) strategies.

Another aspect of generalization relates to the role of *memorization*. For example, is the ability to produce a complex solution dependent on seeing many similar solutions during training, or even on memorizing the solution? Term frequency in the pretraining corpus is known to impact accuracy in simple arithmetic tasks (Razeghi et al., 2022) or factual question answering (Kandpal et al., 2022). On the other hand, Lewkowycz et al. (2022) did not find evidence of memorization in complex outputs, yet their training set and model are not available for inspection. Gaining a full understanding of these factors for complex problems and outputs (e.g., multi-step solutions or proofs) requires more analysis, as well as accessible datasets and models.

7.2 Trustworthy Reasoning

Recent advances in language models have demonstrated their powerful capabilities for mathematical reasoning. However, due to the potential for generating ungrounded answers (Nakano et al., 2021), users can't always trust the predicted outcomes or have to verify then with extra efforts. Even with recent prompting strategies that provide rationales before making predictions (Wei et al., 2022), language models can still hallucinate statements, produce flawed reasoning, and output wrong answers. Consequently, novel approaches that enable more reliable reasoning are needed urgently. Some potential directions for this include: (i) using language models to provide evidence, such as theorems, to support the reasoning process; (ii) incorporating a mechanism that makes a judgment when the model is unsure of the answer; and (iii) using a model itself or another module to detect and locate mistakes in a model's reasoning.

7.3 Learning from Feedback

Another important direction to further improve language models for mathematical reasoning is to let the model learn from feedback. Such a process makes the continual improvement of models' output quality and safety possible. An example is using reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) to align language models with instructions. The idea is to let humans rank the generated outputs of language models and use the learned reward function to finetune the language model with policy gradient (Ouyang et al., 2022; Glaese et al., 2022; Qiu et al., 2022a). In the context of mathematical reasoning, feedback does not necessarily come from humans directly. The outcome of a theorem-proof engine (Jiang et al., 2021; Wu et al., 2021d, 2022c) or the execution result of model-generated scripts can also be used as the reward source (Polu and Sutskever, 2020).

7.4 Multi-modal Mathematical Reasoning

In recent years, there has been growing interest in multi-modal mathematical reasoning, which involves using multiple sources of information, such as text, tables, natural images, and diagrams (Kahou et al., 2018; Kafle et al., 2018; Lu et al., 2021b, 2022b). However, currently available datasets in this domain tend to be small (Zhao et al., 2022), generated from templates (Kahou et al., 2018), or focus on specific topics (Lu et al., 2021a; Chen et al., 2022a). One line of current research involves applying VQA-based frameworks to analyze figures and plots, but this approach can result in significant semantic gaps due to the fact that most VQA models are trained on natural images. One potential direction for future work is to enhance the ability of multi-modal mathematical reasoning systems to tackle more complex and realistic problems. This may involve creating unified models for interpreting and integrating different modalities, as well as developing better evaluation benchmarks to assess the performance of these systems.

8 Conclusion

In this paper, we present a comprehensive survey of deep learning for mathematical reasoning. We review the various tasks, datasets, and deep learning approaches. We also identify several gaps in the existing datasets and methods. Finally, we outline directions for future research and highlight the potential for further exploration in this field. Our goal with this paper is to provide a comprehensive and useful resource for readers interested in the development of deep learning for mathematical reasoning. To aid in this effort, we have created a reading list that will be continually updated in a GitHub repository at https://github.com/lupantech/dl4math.

Limitations

One limitation of our survey work is that it is focused on the intersection of mathematical reasoning and deep learning over the past decade, which may not encompass the entire field and its history. Additionally, our evaluation of existing benchmarks and methods is based on a curated set of papers and may not fully represent the state of the art in the field. Furthermore, due to the fast-paced nature of the field, our survey may not reflect the latest developments and advancements which may have come out close to or after the survey was conducted. Despite these limitations, our survey still provides a valuable overview of the current state and key trends in the field of mathematical reasoning and deep learning, and can serve as a valuable resource for researchers and practitioners working in this field.

Broader Impact

Our survey paper on the intersection of mathematical reasoning and deep learning has the potential to significantly impact the field of artificial intelligence. By providing a comprehensive overview of the key tasks, datasets, and methods that have been developed in the past decade, we give researchers and practitioners a clear understanding of the current state-of-the-art and help them make informed decisions about their own research. Additionally, by evaluating existing benchmarks and methods and discussing future research directions, we aim to identify gaps in the current state of the art and guide future research and development efforts towards more advanced and effective mathematical reasoning systems. Overall, our survey has the potential to contribute to the advancement of mathematical reasoning and deep learning, and have a profound impact on machine learning and natural language processing.

References

Alexander A. Alemi, François Chollet, Niklas Een, Geoffrey Irving, Christian Szegedy, and Josef Urban. 2016. Deepmath - deep sequence models for premise selection. *Advances in neural information processing systems (NeurIPS)*, 29.

Reem Alghamdi, Zhenwen Liang, and Xiangliang Zhang. 2022. Armath: a dataset for solving arabic math word problems. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC)*, pages 351–362.

Chris Alvin, Sumit Gulwani, Rupak Majumdar, and Supratik Mukhopadhyay. 2017. Synthesis of solutions for shaded area geometry problems. In *The Thirtieth International Flairs Conference*.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2357–2367.

Connor Anderson and Ryan Farrell. 2022. Improving fractal pre-training. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1300–1309.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 6077–6086.

Cem Anil, Yuhuai Wu, Anders Johan Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Venkatesh Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. 2022. Exploring length generalization in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*.

Kshitij Bansal, Sarah Loos, Markus Rabe, Christian Szegedy, and Stewart Wilcox. 2019. Holist: An environment for machine learning of higher order logic theorem proving. In *International Conference on Machine Learning (ICML)*, pages 454–463. PMLR.

Bruno Barras, Samuel Boutin, Cristina Cornes, Judicaël Courant, Yann Coscoy, David Delahaye, Daniel de Rauglaudre, Jean-Christophe Filliâtre, Eduardo Giménez, Hugo Herbelin, et al. 1999. The coq proof assistant reference manual. *INRIA, version*, 6(11).

Taylor Berg-Kirkpatrick and Daniel Spokoyny. 2020. An empirical investigation of contextualized number prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4754–4764.

Arindam Bhattacharya. 2017. A survey of question answering for math and science problem. *arXiv preprint arXiv:1705.04530*.

Daniel G Bobrow. 1964. Natural language input for a computer problem solving system. *AI Technical Reports*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901.

Jie Cao and Jing Xiao. 2022. An augmented benchmark dataset for geometric question answering through dual parallel text encoding. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING)*, pages 1511–1520.

Yixuan Cao, Feng Hong, Hongwei Li, and Ping Luo. 2021. A bottom-up dag structure extraction model for math word problems. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 39–46.

François Charton. 2022. Linear algebra with transformers. *Transactions on Machine Learning Research*.

Jiaqi Chen, Tong Li, Jinghui Qin, Pan Lu, Liang Lin, Chongyu Chen, and Xiaodan Liang. 2022a. Unigeo: Unifying geometry logical reasoning via reformulating mathematical expression. In *The 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric Xing, and Liang Lin. 2021a. Geoqa: A geometric question answering benchmark towards multimodal numerical reasoning. In *Findings of the Association for Computational Linguistics (ACL)*, pages 513–523.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021b. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022b. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

Wenhu Chen, Ming Yin, Max Ku, Elaine Wan, Xueguang Ma, Jianyu Xu, Tony Xia, Xinyi Wang, and Pan Lu. 2023. Theoremqa: A theorem-driven question answering dataset. *arXiv preprint arXiv:2305.12524*.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R Routledge, et al. 2021c. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3697–3711.

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022c. Convfinqa: Exploring the chain of numerical reasoning in conversational finance question answering. *arXiv preprint arXiv:2210.03849*.

Ting-Rui Chiang and Yun-Nung Chen. 2019. Semantically-aligned equation generation for solving and reasoning math word problems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2656–2668.

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 1931–1942.

Kyunghyun Cho, Bart van Merrienboer Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. 1996. Automated generation of readable proofs with geometric invariants. *Journal of Automated Reasoning*, 17(3):325–347.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Peter Clark, Oren Etzioni, Tushar Khot, Daniel Khashabi, Bhavana Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, Niket Tandon, et al. 2020. From ‘f’to ‘a’on the ny regents science exams: An overview of the aristo project. *AI Magazine*, 41(4):39–53.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pages 2368–2378.

Edward A Feigenbaum et al. 1963. *Computers and thought*. McGraw-Hill.

Yu Feng, Jing Zhang, Xiaokang Zhang, Lemao Liu, Cuiping Li, and Hong Chen. 2021. Injecting numerical reasoning skills into knowledge base question answering models. *arXiv preprint arXiv:2112.06109*.

Deborah Ferreira and André Freitas. 2020a. Natural language premise selection: Finding supporting statements for mathematical text. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2175–2182.

Deborah Ferreira and André Freitas. 2020b. Premise selection in natural language mathematical texts. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7365–7374.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023. Complexity-based prompting for multi-step reasoning. In *International Conference on Learning Representations (ICLR)*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2022. Pal: Program-aided language models. *arXiv preprint arXiv:2211.10435*.

Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven CH Hoi, Xiaogang Wang, and Hongsheng Li. 2019. Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6639–6648.

Thibault Gauthier, Cezary Kaliszyk, Josef Urban, Ramana Kumar, and Michael Norrish. 2021. TacticToe: Learning to Prove with Tactics. *Journal of Automated Reasoning*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International conference on machine learning (ICML)*, pages 1243–1252. PMLR.

Herbert Gelernter, James R Hansen, and Donald W Loveland. 1960. Empirical explorations of the geometry theorem machine. In *Papers presented at the May 3-5, 1960, western joint IRE-AIEE-ACM computer conference*, pages 143–149.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 946–958.

Kevin Gimpel, Dipanjan Das, and Noah A Smith. 2010. Distributed asynchronous online learning for natural language processing. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 213–222.

Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. 2022. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*.

Adam Grabowski, Artur Korniłowicz, and Adam Naumowicz. 2015. Four decades of mizar. *Journal of Automated Reasoning*, 55(3):191–198.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International Conference on Machine Learning (ICML)*, pages 3929–3938. PMLR.

Jesse Michael Han, Jason Rute, Yuhuai Wu, Edward W Ayers, and Stanislas Polu. 2022. Proof artifact co-training for theorem proving with language models. In *International Conference on Learning Representations (ICLR)*.

Yihan Hao, Mingliang Zhang, Fei Yin, and Linlin Huang. 2022. Pgdp5k: A diagram parsing dataset for plane geometry problems. In *26th International Conference on Pattern Recognition (ICPR)*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. In *35th Conference on Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2744–2751.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Mueller, Francesco Piccinno, and Julian Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4320–4333.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yining Hong, Qing Li, Daniel Ciao, Siyuan Huang, and Song-Chun Zhu. 2021a. Learning by fixing: Solving math word problems with weak supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 4959–4967.

Yining Hong, Qing Li, Ran Gong, Daniel Ciao, Siyuan Huang, and Song-Chun Zhu. 2021b. Smart: A situation model for algebra story problems via attributed grammar. In *AAAI*, pages 13009–13017.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Daniel Huang, Prafulla Dhariwal, Dawn Song, and Ilya Sutskever. 2019. Gamepad: A learning environment for theorem proving. In *International Conference on Learning Representations (ICLR)*.

Danqing Huang, Jing Liu, Chin-Yew Lin, and Jian Yin. 2018. Neural math word problem solver with reinforcement learning. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 213–223.

Danqing Huang, Shuming Shi, Chin-Yew Lin, and Jian Yin. 2017. Learning fine-grained expressions to solve math word problems. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 805–814.

Danqing Huang, Shuming Shi, Chin-Yew Lin, Jian Yin, and Wei-Ying Ma. 2016. How well do computers solve math word problems? large-scale dataset construction and evaluation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 887–896.

Albert Q. Jiang, Sean Welleck, Jin Peng Zhou, Wenda Li, Jiacheng Liu, Mateja Jamnik, Timothée Lacroix, Yuhuai Wu, and Guillaume Lample. 2022a. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs. In *Submitted to The Eleventh International Conference on Learning Representations*.

Albert Qiaochu Jiang, Wenda Li, Jesse Michael Han, and Yuhuai Wu. 2021. Lisa: Language models of isabelle proofs. In *6th Conference on Artificial Intelligence and Theorem Proving (AITP)*.

Albert Qiaochu Jiang, Wenda Li, Szymon Tworkowski, Konrad Czechowski, Tomasz Odrzygóźdź, Piotr Miłoś, Yuhuai Wu, and Mateja Jamnik. 2022b. Thor: Wielding hammers to integrate language models and automated theorem provers. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:8360–8373.

Zhanming Jie, Jierui Li, and Wei Lu. 2022. Learning to reason deductively: Math word problem solving as complex relation extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5944–5955.

Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. Maieutic prompting: Logically consistent reasoning with recursive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1266–1279.

Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. 2018. Dvqa: Understanding data visualizations via question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5648–5656.

Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. 2018. Figureqa: An annotated figure dataset for visual reasoning. In *International Conference on Learning Representations (ICLR)*.

Cezary Kaliszyk, François Chollet, and Christian Szegedy. 2017. Holstep: A machine learning dataset for higher-order logic theorem proving. In *International Conference on Learning Representations (ICLR)*.

Ashwin Kalyan, Abhinav Kumar, Arjun Chandrasekaran, Ashish Sabharwal, and Peter Clark. 2021. How much coffee was consumed during emnlp 2019? fermi problems: A new reasoning challenge for ai. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7318–7328.

Nikhil Kandpal, H. Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2022. Large language models struggle to learn long-tail knowledge. *ArXiv*, abs/2211.08411.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. In *Findings of the Association for Computational Linguistics (EMNLP)*, pages 1896–1907.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*.

Bugeun Kim, Kyung Seo Ki, Donggeon Lee, and Gahgene Gweon. 2020. 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)*, pages 3768–3779.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1571–1581.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 5583–5594.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *36th Conference on Neural Information Processing Systems (NeurIPS)*.

Rik Koncel-K., Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. Mawps: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 1152–1157.

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. 2015. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics (TACL)*, 3:585–597.

Kundan Krishna, Jeffrey Bigham, and Zachary C Lipton. 2021. Does pretraining for summarization require knowledge transfer? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3178–3189.

Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. 2014. Learning to automatically solve algebra word problems. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 271–281.

Guillaume Lample and François Charton. 2020. Deep learning for symbolic mathematics. In *International Conference on Learning Representations (ICLR)*.

Guillaume Lample, Timothee Lacroix, Marie-Anne Lachaux, Aurelien Rodriguez, Amaury Hayat, Thibaut Lavril, Gabriel Ebner, and Xavier Martinet. 2022. Hypertree proof search for neural theorem proving. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:26337–26349.

Yihuai Lan, Lei Wang, Qiyuan Zhang, Yunshi Lan, Bing Tian Dai, Yan Wang, Dongxiang Zhang, and Ee-Peng Lim. 2022. Mwptoolkit: an open-source framework for deep learning-based math word problem solvers. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 13188–13190.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7871–7880.

Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jierui Li, Lei Wang, Jipeng Zhang, Yan Wang, Bing Tian Dai, and Dongxiang Zhang. 2019. Modeling intra-relation in math word problems with different functional multi-head attentions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6162–6167.

Jiwei Li, Alexander H Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2017. Dialogue learning with human-in-the-loop. In *International Conference on Learning Representations (ICLR)*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2020a. What does bert with vision look at? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5265–5275.

Shucheng Li, Lingfei Wu, Shiwei Feng, Fangli Xu, Fengyuan Xu, and Sheng Zhong. 2020b. Graph-to-tree neural networks for learning structured input-output translation with applications to semantic parsing and math word problem. In *Findings of the Association for Computational Linguistics (EMNLP)*, pages 2841–2852.

Wenda Li, Lei Yu, Yuhuai Wu, and Lawrence C Paulson. 2021. Isarstep: a benchmark for high-level mathematical reasoning. In *International Conference on Learning Representations (ICLR)*.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2022a. On the advance of making language models better reasoners. *arXiv preprint arXiv:2206.02336*.

Zhongli Li, Wenxuan Zhang, Chao Yan, Qingyu Zhou, Chao Li, Hongzhi Liu, and Yunbo Cao. 2022b. Seeking patterns, not just memorizing procedures: Contrastive learning for solving math word problems. In *Findings of the Association for Computational Linguistics (ACL)*, pages 2486–2496.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022a. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Percy Liang and Dan Klein. 2009. Online em for unsupervised models. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics (NAACL)*, pages 611–619.

Zhenwen Liang, Jipeng Zhang, Lei Wang, Wei Qin, Yunshi Lan, Jie Shao, and Xiangliang Zhang. 2022b. Mwp-bert: Numeracy-augmented pre-training for math word problem solving. In *Findings of the Association for Computational Linguistics (NAACL)*, pages 997–1009.

Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020. Birds have four legs?! numersense: Probing numerical commonsense knowledge of pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6862–6868.

Xin Lin, Zhenya Huang, Hongke Zhao, Enhong Chen, Qi Liu, Hao Wang, and Shijin Wang. 2021. Hms: A hierarchical solver with dependency-enhanced understanding for math word problem. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 4232–4240.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 158–167.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022a. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022b. TAPEX: Table pre-training via learning a neural SQL executor. In *International Conference on Learning Representations*.

Qianying Liu, Wenyu Guan, Sujian Li, Fei Cheng, Daisuke Kawahara, and Sadao Kurohashi. 2020. Reverse operation based data augmentation for solving math word problems. *IEEE Transactions on Audio, Speech and Language Processing*.

Qianying Liu, Wenyv Guan, Sujian Li, and Daisuke Kawahara. 2019a. Tree-structured decoding for solving math word problems. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 2370–2379.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Sarah Loos, Geoffrey Irving, Christian Szegedy, and Cezary Kaliszyk. 2017. Deep network guided proof search. *arXiv preprint arXiv:1701.06972*.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. 2021a. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. In *The 59th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022a. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023. Chameleon: Plug-and-play compositional reasoning with large language models. *arXiv preprint arXiv:2304.09842*.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2022b. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *International Conference on Learning Representations (ICLR)*.

Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. 2021b. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. In *The 35th Conference on Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022c. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8086–8098.

The mathlib Community. 2020. The lean mathematical library. In *CPP 2020 - Proceedings of the 9th ACM*

SIGPLAN International Conference on Certified Programs and Proofs, co-located with POPL 2020.

Jordan Meadows and Andre Freitas. 2022. A survey in mathematical language processing. *arXiv preprint arXiv:2205.15231*.

Norman D. Megill and David A. Wheeler. 2019. *Metamath: A Computer Language for Mathematical Proofs*. Lulu Press, Morrisville, North Carolina. http://us.metamath.org/downloads/metamath.pdf.

Yuanliang Meng and Anna Rumshisky. 2019. Solving math word problems with double-decoder transformer. *arXiv preprint arXiv:1908.10924*.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing english math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 975–984.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning based text classification: a comprehensive review. *ACM Computing Surveys (CSUR)*, 54(3):1–40.

Swaroop Mishra, Matthew Finlayson, Pan Lu, Leonard Tang, Sean Welleck, Chitta Baral, Tanmay Rajpurohit, Oyvind Tafjord, Ashish Sabharwal, Peter Clark, and Ashwin Kalyan. 2022a. Lila: A unified benchmark for mathematical reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022b. Numglue: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3505–3523.

Eric Mitchell, Joseph J. Noh, Siyan Li, William S. Armstrong, Ananth Agarwal, Patrick Liu, Chelsea Finn, and Christopher D. Manning. 2022. Enhancing self-consistency and performance of pretrained language models with nli. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Leonardo de Moura, Soonho Kong, Jeremy Avigad, Floris van Doorn, and Jakob von Raumer. 2015. The lean theorem prover (system description). In *International Conference on Automated Deduction*, pages 378–388. Springer.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Allen Newell, John Clifford Shaw, and Herbert A Simon. 1957. Empirical explorations of the logic theory machine: A case study in heuristic. In *Proceedings of the Western Joint Computer Conference, IRE-AIEE-ACM 1957*.

Ansong Ni, Jeevana Priya Inala, Chenglong Wang, Oleksandr Polozov, Christopher Meek, Dragomir Radev, and Jianfeng Gao. 2023. Learning from self-sampled correct and partially-correct programs. In *International Conference on Learning Representations (ICLR)*.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HIT)*, pages 2080–2094.

Lawrence C. Paulson. 1994. *Isabelle - A Generic Theorem Prover (with a contribution by T. Nipkow)*, volume 828 of *Lecture Notes in Computer Science*. Springer.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2018. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Stanislas Polu, Jesse Michael Han, Kunhao Zheng, Mantas Baksys, Igor Babuschkin, and Ilya Sutskever. 2023. Formal mathematics statement curriculum learning. In *International Conference on Learning Representations (ICLR)*, volume abs/2202.01344.

Stanislas Polu and Ilya Sutskever. 2020. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*.

Jinghui Qin, Xiaodan Liang, Yining Hong, Jianheng Tang, and Liang Lin. 2021. Neural-symbolic solver for math word problems with auxiliary tasks. In

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL), pages 5870–5881.

Jinghui Qin, Lihui Lin, Xiaodan Liang, Rumin Zhang, and Liang Lin. 2020. Semantically-aligned universal tree-structured solver for math word problems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3780–3789.

Liang Qiu, Yizhou Zhao, Jinchao Li, Pan Lu, Baolin Peng, Jianfeng Gao, and Song-Chun Zhu. 2022a. Valuenet: A new dataset for human value driven dialogue system. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 2468–2484.

Liang Qiu, Yizhou Zhao, Yuan Liang, Pan Lu, Weiyan Shi, Zhou Yu, and Song-chun Zhu. 2022b. Towards socially intelligent agents with mental state transition and human value. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 146–158.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2020. Language models are unsupervised multitask learners. *OpenAI Blog*.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, 21:1–67.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. Equate: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot numerical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 840–854.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems (NeurIPS)*, 28.

Ryokan Ri and Yoshimasa Tsuruoka. 2022. Pretraining with artificial language: Studying transferable knowledge in language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7302–7315.

Benjamin Robaidek, Rik Koncel-Kedziorski, and Hannaneh Hajishirzi. 2018. Data-driven methods for solving algebra word problems. *arXiv preprint arXiv:1804.10718*.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1743–1752.

Subhro Roy and Dan Roth. 2017. Unit dependency graph and its application to arithmetic word problem solving. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Subhro Roy and Dan Roth. 2018. Mapping to declarative knowledge for word problem solving. *Transactions of the Association for Computational Linguistics (TACL)*, 6:159–172.

Subhro Roy, Tim Vieira, and Dan Roth. 2015. Reasoning about quantities in natural language. *Transactions of the Association for Computational Linguistics (TACL)*, 3:1–13.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. *North American Chapter of the Association for Computational Linguistics (NAACL)*.

Mrinmaya Sachan, Kumar Dubey, and Eric Xing. 2017. From textbooks to knowledge: A case study in harvesting axiomatic knowledge from textbooks to solve geometry problems. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 773–784.

Mrinmaya Sachan and Eric Xing. 2017. Learning to solve geometry problems from natural language demonstrations in textbooks. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics*, pages 251–261.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2020. Analysing mathematical reasoning abilities of neural models. In *International Conference on Learning Representations (ICLR)*.

Tal Schuster, Ashwin Kalyan, Alex Polozov, and Adam Tauman Kalai. 2021. Programming puzzles. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual*

Meeting of the Association for Computational Linguistics (ACL), pages 1715–1725.

Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. 2015. Solving geometry problems: Combining text and diagram interpretation. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 1466–1476.

Jianhao Shen, Yichun Yin, Lin Li, Lifeng Shang, Xin Jiang, Ming Zhang, and Qun Liu. 2021. Generate & rank: A multi-task framework for math word problems. In *Findings of the Association for Computational Linguistics (EMNLP)*, pages 2269–2279.

Yibin Shen and Cheqing Jin. 2020. Solving math word problems with multi-encoders and multi-decoders. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, pages 2924–2934.

Shuming Shi, Yuehui Wang, Chin-Yew Lin, Xiaojiang Liu, and Yong Rui. 2015. Automatically solving number word problems by semantic parsing and reasoning. In *Proceedings of the 2015 conference on empirical methods in natural language processing (EMNLP)*, pages 1132–1142.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *36th International Conference on Machine Learning (ICML)*.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. Dream: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics (TACL)*, 7:217–231.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems (NeurIPS)*, 27.

Oyvind Tafjord, Peter Clark, Matt Gardner, Wen-tau Yih, and Ashish Sabharwal. 2019. Quarel: A dataset and models for answering questions about qualitative relationships. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 7063–7071.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL)*, pages 1556–1566.

Avijit Thawani, Jay Pujara, and Ashwin Kalyan. 2022. Estimating numbers without regression. In *36th Conference on Neural Information Processing Systems (NeurIPS 2022) Workshop on MATH-AI*.

Avijit Thawani, Jay Pujara, Pedro A Szekely, and Filip Ilievski. 2021. Representing numbers in nlp: a survey and a vision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HIT)*, pages 644–656.

Shounaak Ughade and Satish Kumbhar. 2019. Survey on mathematical word problem solving using natural language processing. In *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*, pages 1–5. IEEE.

Shyam Upadhyay and Ming-Wei Chang. 2015. Draw: A challenging and diverse algebra word problem set. Technical report, Citeseer.

Shyam Upadhyay and Ming-Wei Chang. 2017. Annotating derivations: A new evaluation strategy and dataset for algebra word problems. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (ACL)*, pages 494–504.

Josef Urban. 2006. Mptp 0.2: Design, implementation, and initial experiments. *Journal of Automated Reasoning*, 37(1):21–43.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do nlp models know numbers? probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5307–5315.

Lei Wang, Yan Wang, Deng Cai, Dongxiang Zhang, and Xiaojiang Liu. 2018a. Translating a math word problem to a expression tree. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1064–1069.

Lei Wang, Dongxiang Zhang, Lianli Gao, Jingkuan Song, Long Guo, and Heng Tao Shen. 2018b. Mathdqn: Solving arithmetic word problems via deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Lei Wang, Dongxiang Zhang, Jipeng Zhang, Xing Xu, Lianli Gao, Bing Tian Dai, and Heng Tao Shen. 2019. Template-based math word problem solvers with recursive neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 7144–7151.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*.

Yan Wang, Xiaojiang Liu, and Shuming Shi. 2017. Deep neural solver for math word problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 845–854.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems (NeurIPS)*.

Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hannaneh Hajishirzi, Yejin Choi, and Kyunghyun Cho. 2021. Naturalproofs: Mathematical theorem proving in natural language. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*.

Sean Welleck, Jiacheng Liu, Ximing Lu, Hannaneh Hajishirzi, and Yejin Choi. 2022a. Naturalprover: Grounded mathematical proof generation with language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2023. Generating sequences by learning to self-correct. In *International Conference on Learning Representations (ICLR)*.

Sean Welleck, Peter West, Jize Cao, and Yejin Choi. 2022b. Symbolic brittleness in sequence models: on systematic generalization in symbolic mathematics. In *AAAI*.

Wu Wen-Tsun. 1986. Basic principles of mechanical theorem proving in elementary geometries. *Journal of automated Reasoning*, 2(3):221–252.

Daniel Whalen. 2016. Holophrasm: a neural automated theorem prover for higher-order logic. *arXiv preprint arXiv:1608.02644*.

Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. 2018. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19.

Qinzhuo Wu, Qi Zhang, Jinlan Fu, and Xuan-Jing Huang. 2020. 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)*, pages 7137–7146.

Qinzhuo Wu, Qi Zhang, and Zhongyu Wei. 2021a. An edge-enhanced hierarchical graph-to-tree network for math word problem solving. In *Findings of the Association for Computational Linguistics (EMNLP)*, pages 1473–1482.

Qinzhuo Wu, Qi Zhang, Zhongyu Wei, and Xuan-Jing Huang. 2021b. Math word problem solving with explicit numerical values. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL)*, pages 5859–5869.

Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. 2022a. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yuhuai Wu, Albert Jiang, Jimmy Ba, and Roger Baker Grosse. 2021c. Int: An inequality benchmark for evaluating generalization in theorem proving. In *International Conference on Learning Representations (ICLR)*.

Yuhuai Wu, Albert Qiaochu Jiang, Wenda Li, Markus Norman Rabe, Charles E Staats, Mateja Jamnik, and Christian Szegedy. 2022b. Autoformalization with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yuhuai Wu, Felix Li, and Percy Liang. 2022c. Insights into pre-training via simpler synthetic tasks. *arXiv preprint arXiv:2206.10139*.

Yuhuai Wu, Markus N Rabe, Wenda Li, Jimmy Ba, Roger B Grosse, and Christian Szegedy. 2021d. Lime: Learning inductive bias for primitives of mathematical reasoning. In *International Conference on Machine Learning (ICML)*, pages 11251–11262. PMLR.

Zhipeng Xie and Shichao Sun. 2019. A goal-driven tree-structured neural model for math word problems. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5299–5305.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning (ICML)*, pages 2048–2057. PMLR.

Kaiyu Yang and Jia Deng. 2019. Learning to prove theorems via interacting with proof assistants. In *International Conference on Machine Learning (ICML)*, pages 6984–6994. PMLR.

Zheng Ye, Shang-Ching Chou, and Xiao-Shan Gao. 2008. An introduction to java geometry expert. In *International workshop on automated deduction in geometry*, pages 189–195. Springer.

Wei Yu, Mengzhu Wang, Xiaodong Wang, Xun Zhou, Yongfu Zha, Yongjian Zhang, Shuyu Miao, and Jingdong Liu. 2021a. Geore: A relation extraction dataset for chinese geometry problems. In *35th Conference on Neural Information Processing Systems (NeurIPS) Workshop on Math AI for Education (MATHAI4ED)*.

Weijiang Yu, Yingpeng Wen, Fudan Zheng, and Nong Xiao. 2021b. Improving math word problems with pre-trained knowledge and hierarchical reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3384–3394.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. In *International Conference on Learning Representations (ICLR)*.

Klim Zaporojets, Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2021. Solving arithmetic word problems by scoring equations with recursive neural networks. *Expert Systems with Applications*, 174:114704.

Dongxiang Zhang, Lei Wang, Luming Zhang, Bing Tian Dai, and Heng Tao Shen. 2019. The gap of semantic parsing: A survey on automatic math word problem solvers. *IEEE transactions on pattern analysis and machine intelligence*, 42(9):2287–2305.

Jipeng Zhang, Roy Ka-Wei Lee, Ee-Peng Lim, Wei Qin, Lei Wang, Jie Shao, and Qianru Sun. 2020a. Teacher-student networks with multiple decoders for solving math word problem. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Jipeng Zhang, Lei Wang, Roy Ka-Wei Lee, Yi Bin, Yan Wang, Jie Shao, and Ee-Peng Lim. 2020b. Graph-to-tree learning for solving math word problems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3928–3937.

Ming-Liang Zhang, Fei Yin, Yi-Han Hao, and Cheng-Lin Liu. 2022. Learning to understand plane geometry diagram. In *36th Conference on Neural Information Processing Systems (NeurIPS) Workshop on MATH-AI*.

Qiyuan Zhang, Lei Wang, Sicheng Yu, Shuohang Wang, Yang Wang, Jing Jiang, and Ee-Peng Lim. 2021. Noahqa: Numerical reasoning with interpretable graph question answering dataset. In *Findings of the Association for Computational Linguistics (EMNLP)*, pages 4147–4161.

Wenhe Zhang, Chi Zhang, Yixin Zhu, and Song-Chun Zhu. 2020c. Machine number sense: A dataset of visual arithmetic problems for abstract and relational reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 1332–1340.

Xikun Zhang, Deepak Ramachandran, Ian Tenney, Yanai Elazar, and Dan Roth. 2020d. Do language embeddings capture scales? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 292–299.

Xikun Zhang, Deepak Ramachandran, Ian Tenney, Yanai Elazar, and Dan Roth. 2020e. Do language embeddings capture scales? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 292–299.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020f. Dialogpt: Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *International Conference on Learning Representations (ICLR)*.

Wei Zhao, Mingyue Shang, Yang Liu, Liang Wang, and Jingming Liu. 2020. Ape210k: A large-scale and template-rich dataset of math word problems. *arXiv preprint arXiv:2009.11506*.

Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. 2022. Multihiertt: Numerical reasoning over multi hierarchical tabular and textual data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6588–6600.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning (ICML)*, pages 12697–12706. PMLR.

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. 2022. Minif2f: a cross-system benchmark for formal olympiad-level mathematics. In *International Conference on Learning Representations (ICLR)*.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "Going on a vacation" takes longer than "Going for a walk": A Study of Temporal Commonsense Understanding. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *International Conference on Learning Representations (ICLR)*.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-JCNLP)*, pages 3277–3287.

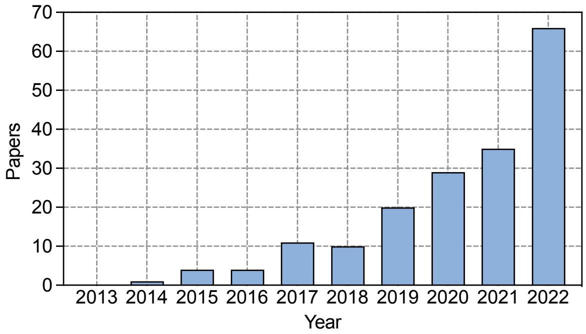


Figure 3: Estimated counts of annually published papers on deep learning for mathematical reasoning. This field has been experiencing rapid growth since 2018.

A Mathematical Reasoning Datasets

In this section, we will examine the various datasets currently available for the study of mathematical reasoning using deep learning methods. A summary of the commonly used datasets in this field can be found in Table 7.

A.1 Math Word Problem Solving

Developing algorithms to solve math word problems (MWPs) automatically has been an interest of NLP researchers for decades (Feigenbaum et al., 1963; Bobrow, 1964). A math word problem (also termed an algebraic or arithmetic word problem) describes a brief narrative that involves characters, entities, and quantities. The mathematical relationship of an MWP can be modeled with a set of equations whose solution reveals the final answer to the question. A typical example is shown in Table 1. A question involves the four basic arithmetic operations of addition, subtraction, multiplication, and division with single or multiple operation steps. The challenge of MWPs for NLP systems lies in the need for language comprehension, semantic parsing, and multiple mathematical reasoning skills.

Existing MWP datasets cover grade school problems, which are crawled from online learning websites (Koncel-Kedziorski et al., 2015), collected from textbooks, or manually annotated by human workers (Patel et al., 2021). Early math word problem datasets are relatively small or limited to a small number of operation steps (Hosseini et al., 2014; Kushman et al., 2014; Roy et al., 2015). Some recently curated datasets aim to increase problem diversity and difficulty levels. For example, Ape210K (Zhao et al., 2020) consists of 210k elementary math word problems, which is the largest publicly available. The problems in GSM8K (Cobbe et al., 2021) can involve up to 8 steps to solve. SVAMP (Patel et al., 2021) is a benchmark that tests the robustness of deep learning models to math word problems with simple variations. More recently built datasets involve modalities beyond text. For example, IconQA (Lu et al., 2021b) provides an abstract diagram as a visual context, while TabMWP (Lu et al., 2022b) provides a tabular context for each problem.

Most MWP datasets provide annotated equations as a rationale for the solution (e.g., Table 1). To improve the performance and interpretability of the learned solvers, MathQA (Tafjord et al., 2019) is annotated with precise operation programs, and MathQA-Python (Austin et al., 2021) is provided with specific Python programs instead. Another line of datasets annotates the problems with multi-step natural language solutions that are regarded as more human-readable (Ling et al., 2017; Cobbe et al., 2021; Lu et al., 2022b). Lila (Mishra et al., 2022a) annotates many of the previously mentioned MWP datasets with Python program rationales.

A.2 Theorem Proving

Recently, there has been increased interest in using language models for theorem proving in formal *interactive theorem provers* (ITP) (e.g., Polu and Sutskever (2020); Han et al. (2022); Polu et al. (2023); Jiang et al. (2022b,a); Lample et al. (2022)). Example ITPs include Lean (Moura et al., 2015), Isabelle (Paulson, 1994), Coq (Barras et al., 1999), and Metamath (Megill and Wheeler, 2019). To prove a theorem in an ITP, the theorem is stated in the ITP's programming language, then simplified by generating "proof steps" until it is reduced to known facts. The result is a sequence of steps that constitutes a verified proof.

Data sources for neural theorem proving in ITPs include interactive learning environments that interface with ITPs, and datasets derived from proofs in ITP libraries. For example, CoqGym (Yang and Deng, 2019) provides an interactive environment and 71K human-written proofs for the Coq ITP. For Isabelle, PISA (Jiang et al., 2021) enables interaction and provides a dataset of 183k proofs mined from the Isabelle standard library and Archive of Formal Proofs. For Lean, LeanStep (Han et al., 2022) provides a dataset of proof-steps from Lean's mathematical library along with auxiliary tasks, while Lean-Gym (Polu et al., 2023) provides an interactive REPL. The miniF2F (Zheng et al., 2022) benchmark aims to provide a shared benchmark across ITPs, consisting of 488 problem statements sourced from mathematical competitions.

Other resources provide proxy environments or tasks. For example, INT (Wu et al., 2021c) provide a synthetic proving environment to measure six different types of generalization. Li et al. construct IsarStep using the Isabelle Archive of Formal Proofs, and propose a task of filling in a missing intermediate proposition. Early applications of deep learning for formal theorem proving focus on selecting relevant premises (Alemi et al., 2016).

Informal theorem proving presents an alternative medium for theorem proving, in which statements and proofs are written in the mixture of natural language and symbols used in "standard" mathematics (e.g., in LaTeX), and are checked for correctness by humans. Early work focuses on selecting relevant premises (Ferreira and Freitas, 2020b,a). Welleck et al. (2021) develop NaturalProofs, a large-scale dataset of 32k informal mathematical theorems, definitions, and proofs, and provide a benchmark for premise selection via retrieval and generation tasks. Welleck et al. (2022a) adapt NaturalProofs for full proof generation, and provide a human evaluation protocol and proxy automatic metrics.

An emerging area of research aims to combine elements of informal and formal theorem proving. For example, Wu et al. (2022b) explore translating informal statements into formal statements, while Jiang et al. (2022a) release a new version of the miniF2F benchmark augmented with informal statements and proofs, which we refer to as *miniF2F+informal*. Jiang et al. (2022a) explore translating provided (or generated) informal proofs into formal proofs.

A.3 Geometry Problem Solving

Automated geometry problem solving (GPS) is also a long-standing AI task in mathematical reasoning research (Gelernter et al., 1960; Wen-Tsun, 1986; Chou et al., 1996; Ye et al., 2008) and has attracted much attention in recent years. Different from a math word problem, a geometry problem consists of a textual description in natural language and a geometric diagram. As shown in Figure 2, the multimodal inputs describe the entities, attributes, and relationships of geometric elements, and the goal is to find the numeric solution to an unknown variable. GPS is a challenging task for deep learning methods due to the complex skills it requires. It involves the ability to parse multimodal information, perform symbolic abstraction, utilize theorem knowledge, and conduct quantitative reasoning.

Some early datasets are proposed to facilitate research in this domain (Seo et al., 2015; Alvin et al., 2017; Sachan et al., 2017; Sachan and Xing, 2017). However, these datasets are relatively small or not publicly available, which limits the development of deep learning methods. In response to this limitation, Lu et al. create the Geometry3K dataset, which consists of 3,002 multi-choice geometry problems with unified logic form annotations for the multimodal inputs. More recently, larger-scale datasets such as GeoQA (Chen et al., 2021a), GeoQA+ (Cao and Xiao, 2022), and UniGeo (Chen et al., 2022a) have been introduced and are annotated with programs that can be learned by neural solvers and executed to obtain the final answers.

A.4 Math Question Answering

Numerical reasoning is a core ability within human intelligence and plays an important role in many NLP tasks. Aside from theorem proving and grade-level math word problem solving, there is a wide range of question answering (QA) benchmarks that center around mathematical reasoning. In this work, we refer to these tasks as math question answering (MathQA). A large number of datasets have been presented recently. For example, QuaRel (Tafjord et al., 2019) is a dataset of diverse story questions that involve 19 different types of quantities. McTaco (Zhou et al., 2019) studies temporal commonsense problems, while Fermi (Kalyan et al., 2021) studies Fermi problems whose answers can only be approximately estimated.

Recent studies have shown that state-of-the-art mathematical reasoning systems might suffer from brittleness in reasoning, in that the models rely on spurious signals and plug-and-chug calculations in the specific dataset to achieve "satisfactory" performance (Hendrycks et al., 2021b; Mishra et al., 2022b). To address this issue, new benchmarks are proposed from various aspects. The Mathematics dataset (Saxton et al., 2020) consists of many different types of mathematics problems, covering arithmetic, algebra, probability, and calculus. The dataset allows for measuring the algebraic generalization ability of a model. Similarly, MATH (Hendrycks et al., 2021b) consists of challenging competition mathematics to measure the problem-solving ability of models in complex scenarios.

Some work incorporates tabular contexts in the question inputs. For example, FinQA (Chen et al., 2021c), TAT-QA (Zhu et al., 2021), and MultiHiertt (Zhao et al., 2022) collect questions that require both table understanding and numeric reasoning to answer. Others, instead, present large-scale unified benchmarks for mathematical reasoning (Mishra et al., 2022b,a; Chen et al., 2023). NumGLUE (Mishra et al., 2022b) is a multi-task benchmark with the goal of evaluating the performance of models on eight different tasks. Mishra et al. 2022a push this direction further and presents Lila, which consists of 23 mathematical reasoning tasks, spanning a wide range of mathematics topics, linguistic complexity, question formats, and background knowledge requirements.

A.5 Other Quantitative Problems

Numbers are an integral part of our daily lives, and we humans reason with numbers in a variety of tasks, such as understanding news, reports, elections, and markets. This has led many in the community to question whether AI systems can effectively perform quantitative reasoning in everyday scenarios. To this end, various benchmarks have been developed to evaluate the capabilities of AI systems in this area.

Diagrams, such as figures, charts, and plots, are essential media that convey large amounts of information in a concise way. FigureQA (Kahou et al., 2018), DVQA (Kafle et al., 2018), MNS (Zhang et al., 2020c), PGDP5K (Hao et al., 2022), and GeoRE (Yu et al., 2021a), are released to investigate models' abilities to reason about quantitative relationships among entities grounded in diagrams. NumerSense (Lin et al., 2020), instead, examines whether and to what extent existing pre-trained language models can induce numerical commonsense knowledge. EQUATE (Ravichander et al., 2019) formalizes aspects of quantitative reasoning in a natural language inference framework. Quantitative reasoning can appear frequently in specific domains like finance, science, and programming. For instance, the ConvFinQA (Chen et al., 2022c) targets numerical reasoning over financial reports in a conversational question answering format. ScienceQA (Lu et al., 2022a) involves numerical reasoning in scientific domains, while P3 (Schuster et al., 2021) studies the function inference ability of deep learning models to find a valid input which makes the given program return `True`.

Dataset	Task	Size	Input	Output	Rationale	Domain
Verb395 (2014)	MWP	395	Question	Number	Equation	Math
Alg514 (2014)	MWP	514	Question	Number	Equation	Math
IL (2015)	MWP	-	Question	Number	Equation	Math
SingleEQ (2015)	MWP	508	Question	Number	Equation	Math
DRAW (2015)	MWP	1,000	Question	Number	Equation	Math
Dolphin1878 (2015)	MWP	1,878	Question	Number	Equation	Math
Dolphin18K (2016)	MWP	18,460	Question	Number	Equation	Math
MAWPS (2016)	MWP	3,320	Question	Number	Equation	Math
AllArith (2017)	MWP	831	Question	Number	Equation	Math
DRAW-1K (2017)	MWP	1,000	Question	Number	Equation	Math
Math23K (2017)	MWP	23,162	Question	Number	Equation	Math
AQuA (2017)	MWP	100,000	Question	Option	Natural language	Math
Aggregate (2018)	MWP	1,492	Question	Number	Equation	Math
MathQA (2019)	MWP	37,297	Question	Number	Program	Math
ASDiv (2020)	MWP	2,305	Question	Number	Equation	Math
HMWP (2020)	MWP	5,470	Question	Number	Equation	Math
Ape210K (2020)	MWP	210,488	Question	Number	Equation	Math
SVAMP (2021)	MWP	1,000	Question	Number	Equation	Math
GSM8K (2021)	MWP	8,792	Question	Number	Natural language	Math
IconQA (2021b)	MWP	107,439	Figure+Question	Option+Text span	✗	Math
MathQA-Python (2021)	MWP	23,914	Question	Number	Python program	Math
ArMATH (2022)	MWP	6,000	Question	Number	Equation	Math
TabMWP (2022b)	MWP	38,431	Table+Question	Option+Number	Natural language	Math
MML (2015)	TP	57,882	Statement	Proof steps	✗	Math
HolStep (2017)	TP	2,209,076	Statement	Proof steps	✗	Math
Gamepad (2019)	TP	-	Statement	Proof steps	✗	Math
CoqGym (2019)	TP	71,000	Statement	Proof steps	✗	Math
HOList (2019)	TP	29,462	Statement	Proof steps	✗	Math
IsarStep (2021)	TP	860,000	Statement	Proof steps	✗	Math
PISA (2021)	TP	183,000	Statement	Proof steps	✗	Math
INT (2021c)	TP	-	Statement	Proof steps	✗	Math
NaturalProofs (2021)	TP	32,000	Statement	Proof steps	✗	Math
NaturalProofs-Gen (2022a)	TP	14,500	Statement	Proof steps	✗	Math
miniF2F (2022)	TP	488	Statement	Proof steps	✗	Math
miniF2F+informal (2022a)	TP	488	Statement	Proof steps	✗	Math
LeanStep (2022)	TP	21,606,000	Statement	Proof steps	✗	Math
GEOS (2015)	GPS	186	Figure+Question	Option	✗	Geometry
GeoShader (2017)	GPS	102	Figure+Question	Number	✗	Geometry
GEOS++ (2017)	GPS	1,406	Figure+Question	Number	✗	Geometry
GEOS-OS (2017)	GPS	2,235	Figure+Question	Option	Demonstration	Geometry
Geometry3K (2021a)	GPS	3,002	Figure+Question	Option	Logical form	Geometry
GeoQA (2021a)	GPS	4,998	Figure+Question	Option	Program	Geometry
GeoQA+ (2022)	GPS	12,054	Figure+Question	Option	Program	Geometry
UniGeo (2022a)	GPS/TP	14,541	Figure+Question	Option	Program	Geometry
Quarel (2019)	MathQA	2,771	Question	Option	Logical form	Math
McTaco (2019)	MathQA	13,225	Text+Question	Option	✗	Time
DROP (2019)	MathQA	96,567	Passage+Question	Number+Text span	✗	Math
Mathematics (2020)	MathQA	2,010,000	Question	Free-form	Number	Math
FinQA (2021c)	MathQA	8,281	Text+Table+Q	Number	Program	Finance
Fermi (2021)	MathQA	11,000	Question	Number	Program+Fact	Math
MATH (2021b)	MathQA	12,500	Question	Number	Natural language	Math
TAT-QA (2021)	MathQA	16,552	Text+Table+Q	Number+Text span	✗	Finance
AMPS (2021b)	MathQA	5,000,000	Question	-	$\LaTeX$	Math
MultiHiertt (2022)	MathQA	10,440	Text+Table+Q	Number+Text span	Expression	Finance
NumGLUE (2022b)	MathQA	101,835	Text+Question	Number+Text span	✗	Math
Lila (2022a)	MathQA	134,000	Text+Question	Free-form	Python program	Math
FigureQA (2018)	VQA	1,000,000+	Figure+Question	Binary	✗	Math
DVQA (2018)	VQA	3,487,194	Figure+Question	Text span	Number+Text span	Math
DREAM (2019)	ConvQA	10,197	Dialog+Question	Option	✗	Math
EQUATE (2019)	NLI	-	Premise+Hypothesis	Binary	✗	Math
NumerSense (2020)	Filling	13,600	Masked question	Word	✗	Math
MNS (2020c)	IQ Test	-	Figure	Number	✗	Math
P3 (2021)	Puzzle	397	Text	Program	✗	Math
NOAHQA (2021)	ConvQA	21,347	Dialog+Question	Text span	Reasoning graph	Math
ConvFinQA (2022c)	ConvQA	3,892	Report+Dialog+Q	Number	Expression	Math
PGDP5K (2022)	Parsing	5,000	Figure+Question	Number	✗	Geometry
GeoRE (2022a)	Parsing	12,901	Figure+Question	Number	✗	Geometry
ScienceQA (2022a)	VQA	21,208	Context+Question	Option	Natural language	Science

Table 7: A summarization of mathematical reasoning datasets.

Paper	Task	Problem	Network	Encod	Decod	ATT	Description
DNS (Wang et al., 2017)	MWP	Generation	Seq2Seq	GRU	LSTM	✗	The first deep MWP solver
AnsRat (Ling et al., 2017)	MWP	Generation	Seq2Seq	LSTM	LSTM	✗	Trained with staged back-propagation
Math-EN (Wang et al., 2018a)	MWP	Generation	Seq2Seq	BiLSTM	LSTM	✔	A standard Seq2Seq model with attention
CASS (Huang et al., 2018)	MWP	Generation	Seq2Seq	BiGRU	BiGRU	✔	Copy and alignment with RL
S-Aligned (Chiang and Chen, 2019)	MWP	Generation	Seq2Seq	BiLSTM	LSTM	✔	Operating symbols
T-RNN (Wang et al., 2019)	MWP	Generation	Seq2Seq	BiLSTM	BiLSTM	✔	Predicting a tree-structure math template
GROUP-ATT (Li et al., 2019)	MWP	Generation	Seq2Seq	BiLSTM	LSTM	✔	Group attention
SMART (Hong et al., 2021b)	MWP	Generation	Seq2Seq	-	-	✗	Explicitly incorporating values
SelfAtt (Robaidek et al., 2018)	GPS	Classification	Seq2Seq	BiLSTM	-	✔	Multi-hop self-attention
QuaSP+ (Tafjord et al., 2019)	MathQA	Generation	Seq2Seq	BiLSTM	LSTM	✗	Adopting attributed grammar
AST-Dec (Liu et al., 2019a)	MWP	Generation	Seq2Tree	BiLSTM	Tree	✔	Using prefix order decoding
GTS (Xie and Sun, 2019)	MWP	Generation	Seq2Tree	BiGRU	Tree	✔	A goal-driven tree-structured approach
KA-S2T (Wu et al., 2020)	MWP	Generation	Seq2Tree	BiLSTM	Tree	✔	A knowledge-aware method
TSN-MD (Zhang et al., 2020a)	MWP	Generation	Seq2Tree	BiGRU	Tree	✔	A teacher-student network
T-LSTM (Zaporojets et al., 2021)	MWP	Generation	Seq2Tree	BiLSTM	Tree	✗	A child-sum tree-LSTM model
NT-LSTM (Zaporojets et al., 2021)	MWP	Generation	Seq2Tree	BiLSTM	Tree	✗	An N-ary tree-LSTM model
NS-Solver (Qin et al., 2021)	MWP	Generation	Seq2Tree	BiGRU	Tree	✔	A neural-symbolic solver with programs
NumS2T (Wu et al., 2021b)	MWP	Generation	Seq2Tree	BiLSTM	Tree	✔	Explicitly incorporating values
HMS (Lin et al., 2021)	MWP	Generation	Seq2Tree	GRU	Tree	✔	A word-clause-problem encoder
LBF (Hong et al., 2021a)	MWP	Generation	Seq2Tree	BiGRU	Tree	✔	A learning-by-fixing (LBF) framework
Seq2DAG (Cao et al., 2021)	MWP	Generation	Seq2Graph	GRU	Graph	✗	A direct acyclic graph (DAG) structure
Graph2Tree (Zhang et al., 2020b)	MWP	Generation	Graph2Tree	Graph	Tree	✗	Generating better solution expressions
Multi-E/D (Shen and Jin, 2020)	MWP	Generation	Graph2Tree	Graph	Tree	✔	A graph encoder and a tree-bad decoder
Graph2Tree (Li et al., 2020b)	MWP	Generation	Graph2Tree	Graph	Tree	✔	A graph-to-tree neural network
EEH-G2T (Wu et al., 2021a)	MWP	Generation	Graph2Tree	Graph	Tree	✗	A hierarchical graph-to-tree model
ASTactic (Yang and Deng, 2019)	TP	Generation	Tree2Seq	TreeLSTM	GRU	✔	Generating tactics as programs
MathDQN (Wang et al., 2018b)	MWP	Search	DQN	-	-	✗	RL with a deep Q-network (DQN)
DDT (Meng and Rumshisky, 2019)	MWP	Generation	Transformer	Trm	Trm	✔	A Transformer-based model
DeepMath (Alemi et al., 2016)	TP	Classification	CNN	CNN	-	✗	The first deep large scale theorem prover
Holophrasm (Whalen, 2016)	TP	Classification	BiGRU	BiGRU	-	✗	A neural prover for higher-order logic
CNNTP (Loos et al., 2017)	TP	Classification	CNN	CNN	-	✗	A CNN-based theorem prover
WaveNetTP (Loos et al., 2017)	TP	Classification	WaveNet	WaveNet	-	✗	A WaveNet-based theorem prover
DeepHOL (Bansal et al., 2019)	TP	Generation	WaveNet	WaveNet	-	✗	A neural theorem prover with RL
NGS (Chen et al., 2021a)	GPS	Generation	VQA	LSTM*	LSTM	✔	The first deep geometry solver
PGDPNet (Zhang et al., 2022)	Parsing	Generation	GNN	-	-	✗	A neural diagram parser with GNN

Table 8: A summarization of deep neural network models for mathematical reasoning. **Encod**: encoder, **Decod**: decoder, **ATT**: Attention. LSTM*: ResNet + LSTM, Trm: Transformer

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations Section on page 10.

☑ A2. Did you discuss any potential risks of your work?
Limitations Section on page 10.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
Not applicable. Left blank.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Not applicable. Left blank.

C ☒ Did you run computational experiments?

Left blank.

□ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Not applicable. Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

□ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Not applicable. Left blank.

□ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Not applicable. Left blank.

□ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.