Introduction to Mathematical Language Processing: Informal Proofs, Word Problems, and Supporting Tasks

Jordan Meadows[1] and **André Freitas**[1,2]
[1]Department of Computer Science, University of Manchester, UK
[2]Idiap Research Institute, Switzerland
jordan.meadows@postgrad.manchester.ac.uk
andre.freitas@idiap.ch

Abstract

Automating discovery in mathematics and science will require sophisticated methods of information extraction and abstract reasoning, including models that can convincingly process relationships between mathematical elements and natural language, to produce problem solutions of real-world value. We analyze mathematical language processing methods across five strategic sub-areas (identifier-definition extraction, formula retrieval, natural language premise selection, math word problem solving, and informal theorem proving) from recent years, highlighting prevailing methodologies, existing limitations, overarching trends, and promising avenues for future research.

1 Introduction

> *Prove that there is no function f from the set of non-negative integers into itself such that $f(f(n)) = n + 1987$ for every n.*
>
> *Show that the nearest neighbor interaction Hamiltonian of an electronic quasiparticle in Graphene can be written as $\mathcal{H} = \hbar\Omega \sum_{\boldsymbol{q}} (f_{\boldsymbol{q}} b^{\dagger}_{\boldsymbol{q}} a_{\boldsymbol{q}} + f^{*}_{\boldsymbol{q}} a^{\dagger}_{\boldsymbol{q}} b_{\boldsymbol{q}})$.*
>
> *How is the sun's atmosphere hotter than its surface?*

If we hope to use machines to derive *mathematically rigorous* and explainable solutions to address such questions, models must reason over both natural language and mathematical elements such as equations, expressions, and variables. Given some input problem description, the ideal model is at least capable of recalling relevant statements *(premise selection)*, assigning contextual descriptions to math elements within that text *(identifier-definition extraction)*, and performing robust manipulation of equations and expressions towards an explainable reasoning argument *(informal theorem proving)*. Previous years have advanced many of the components required to deliver this vision. Transformer-based (Vaswani et al., 2017), large language models (LLMs) (Brown et al., 2020; Chen et al., 2021) have begun to exhibit mathematical (Rabe et al., 2020) and logical (Clark et al., 2020) capabilities. Graph-based models also show competence in premise selection (Ferreira and Freitas, 2020b), math question answering (Feng et al., 2021), and math word problems (MWPs) (Zhang et al., 2022b). The evolutionary path of mathematical language processing can be traced from MWPs (Feigenbaum and Feldman, 1963; Bobrow, 1964; Charniak, 1969) and linguistic analysis of formal proofs (Zinn, 1999, 2003), to the present day, where transformers and graph-based models deliver leading metrics in math and language reasoning tasks, complemented by symbolic methods (Zhong et al., 2022). This survey provides a synthesis of this recent evolutionary arch: We consider five representative tasks with examples, describe contributions leading to the current state-of-the-art, discuss notable limitations of the current solutions, overarching trends, and promising research directions.

2 Representative Tasks

There is an abundance of tasks considering mathematical language, such as question answering (Hopkins et al., 2019; Feng et al., 2021; Lewkowycz et al., 2022; Mansouri et al., 2022b) and headline generation (Yuan et al., 2020; Peng et al., 2021). *Mathematical language processing* (MLP) itself has been described in the context of

Transactions of the Association for Computational Linguistics, vol. 11, pp. 1162–1184, 2023. https://doi.org/10.1162/tacl_a_00594
Action Editor: David Chiang. Submission batch: 2/2023; Revision batch: 5/2023; Published 9/2023.

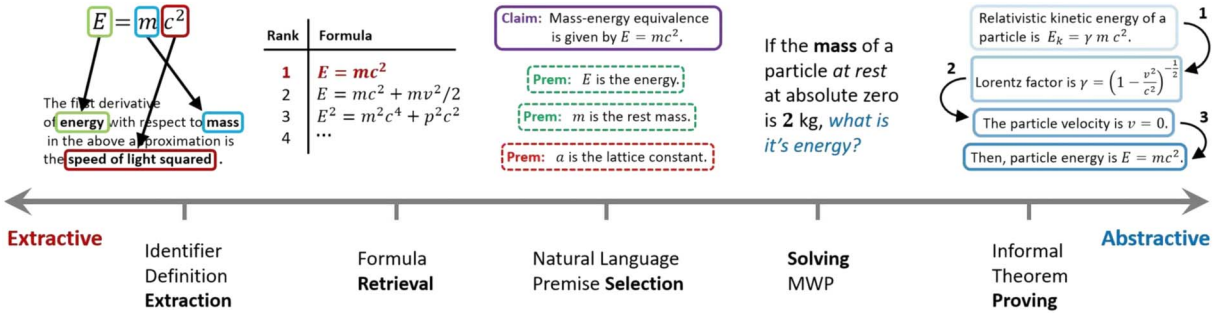

Figure 1: *Extractive* tasks tend to not require inference chains to solve them, compared to more *abstractive* tasks. *Identifier-definition extraction* assigns identifiers $(e.g., \psi(x))$ to their context. *Formula retrieval* considers the structure of formulae, and scores them based on similarity to a query formula. *Premise selection* selects statements most likely to be useful for solving a proof. *Solving MWPs* (math word problems) involves calculating solutions to arithmetic problems. *Informal theorem proving* involves the production of proofs and inference chains combining natural and mathematical language.

various targeted texts, such as linking variables to descriptions (Pagael and Schubotz, 2014), grading answers (Lan et al., 2015), and deriving abstract representations for downstream applications (Wang et al., 2021). We take an inclusive stance, selecting a few choice tasks spanning surface-level retrieval, as seen in *identifier-definition extraction* and *formula retrieval* tasks, through models which require the encoding of formal abstractions and implicit reasoning chains, such as *solving MWPs* and *informal theorem proving*. These areas are projected onto an inference spectrum displayed in Figure 1. Extractive tasks are positioned to the surface form of the text (information retrieval perspective), including identification of relevant mathematical statements, ranking lists of formulae, and linking variables to contextual definitions. Logical puzzle solvers (Groza and Nitu, 2022) and informal reasoning generation models (Lewkowycz et al., 2022) exist far into the abstractive side, due to the *step-wise* and sometimes symbolic reasoning required to address them. The use of ''formal'' versus ''informal'' differentiates strict automated theorem prover (ATP) approaches requiring the use of a consistent formal language representation (Rudnicki, 1992) and hard-coded logic (Bansal et al., 2019), from approaches that input mathematical language and infer without necessary reliance on strict symbolic and logical inference mechanisms. Autoformalization (Szegedy, 2020; Wu et al., 2022) aims to cross this divide. We consider informal methods for solving five representative tasks in this context, with examples given below, visually displayed in Figure 1.

Identifier-Definition Extraction. The assignment of meaning to otherwise vague mathematical elements. Without context, equations such as $\mathbf{p} = \hbar\mathbf{k}$ are ambiguous. *What meaning is attributed to* $\mathbf{k}$? This task involves finding (identifier, definition) pairs, such as $(\mathbf{k}$, wavevector$)$ (Kristianto et al., 2012; Stathopoulos et al., 2018).

Formula Retrieval. Mathematical language includes math elements written in markup languages such as LaTeX. Given a query formula, the Wikipedia Formula Browsing task (Zanibbi et al., 2016a; Mansouri et al., 2022b) involves ranking a list of candidate formulae in terms of their similarity to that formula. For example, given the query $x^2 + y^2 = r^2$, the formula $a^2 + b^2 = c^2$ should rank higher than $y = mx + c$.

Natural Language Premise Selection (NLPS). Given a mathematical statement s that requires proof, and a collection of premises P, this task consists of retrieving the premises in P that are most likely to be useful for proving s (Ferreira and Freitas, 2020a; Valentino et al., 2022). For example, given the purple claim statement in Figure 1, a NLPS model should select the green statements as premises, excluding the red.

Math Word Problem Solving. Solving arithmetic (Roy and Roth, 2016) or algebra (Kushman et al., 2014) word problems. *Andrew has 3 dogs. If they each give birth to 2 others, how many dogs will he have?* An example requiring premise selection and identifier-definition extraction is given in Figure 1.

Informal Theorem Proving. Outputting reasoning chains from premises in order to ''prove'' a mathematical language statement. From Figure 1, the energy of the particle is $E_k = \gamma mc^2$. Substituting $v = 0$ into the Lorentz factor gives $\gamma = 1$, and substituting $\gamma = 1$ into $E_k = \gamma mc^2$ gives $E_k = mc^2$. Such informal reasoning does not rely on formal frameworks, such as Fitch-style proofs, to infer quantitative results (Lewkowycz et al., 2022).

3 Methods

We highlight key points abstracted from task approaches in bold, give an overview of methods in Table 1, and discuss approach specific limitations in the Appendix.

3.1 Identifier-Definition Extraction

A significant proportion of variables or identifiers in formulae or text are explicitly defined within a discourse context (Wolska and Grigore, 2010). Descriptions are usually local to the first instance of the identifiers in the discourse. It is the broad goal of identifier-definition extraction and related tasks to pair-up variables with their intended meaning.

The task has not converged to a canonical form. Despite the clarity of its overall aim, the task has materialized into different forms: Kristianto et al. (2012) predict descriptions given *expressions*, Pagael and Schubotz (2014) predict descriptions given identifiers through *identifier-definition extraction*, Stathopoulos et al. (2018) predict if a type matches a variable through *variable typing*, and Jo et al. (2021) predict notation given context through *notation auto-suggestion* and *notation consistency checking* tasks. More concretely, *identifier-definition extraction* (Schubotz et al., 2016a) involves *scoring* identifier-definiens pairs, where a definiens is a potential natural language description of the identifier. Given graph nodes from predefined variables V and types T, *variable typing* (Stathopoulos et al., 2018) is the task of *classifying* whether edges $V \times T$ are either existent (positive) or non-existent (negative), where a positive classification means a variable matches with the type. *Notation auto-suggestion* (Jo et al., 2021) uses the text of both the sentence containing notation and the previous sentence to *model* future notation from the vocabulary of the tokenizer. This area can be traced from an early ranking task (Pagael and Schubotz, 2014) reliant on heuristics and rules (Alexeeva et al., 2020), through ML-based edge classification (Stathopoulos et al., 2018), to language modeling with Transformers (Jo et al., 2021). Different datasets are proposed for each task variant.

There is a high variability in scoping definitions. The scope from which identifiers are linked to descriptions varies significantly, and it is difficult to compare model performance even when tackling the same variant of the task (Schubotz et al., 2017; Alexeeva et al., 2020). At a local context, models such as Pagael and Schubotz (2014) and Alexeeva et al. (2020) match identifiers with definitions from the same document ''as the author intended'', while other identifier-definition extraction methods (Schubotz et al., 2016a, 2017) rely on data external to a given document, such as links to semantic concepts on Wikidata and NTCIR-11 test data (Schubotz et al., 2015). At a broader context, the variable typing model proposed in Stathopoulos et al. (2018) relies on an external dictionary of types (Stathopoulos and Teufel, 2015; Stathopoulos and Teufel, 2016; Stathopoulos et al., 2018) extracted from both the Encyclopedia of Mathematics[1] and Wikipedia.

Vector representations have evolved to transfer knowledge from previous tasks, allowing downstream variable typing tasks to benefit from pre-trained embeddings. Overall, vector representations of text have evolved from feature-based vectors learned from scratch for a single purpose, to the modern paradigm of pre-trained embeddings re-purposed for novel tasks. Kristianto et al. (2012) input pattern features into a conditional random fields model for the purpose of identifying definitions of expressions in LaTeX papers while Kristianto et al. (2014a) learn vectors through a linear-kernel SVM with input features comprising of sentence patterns, part-of-speech (POS) tags, and tree structures. Stathopoulos et al. (2018) extend this approach by adding type- and variable-centric features as a baseline also with a linear kernel. Alternatively, Schubotz et al. (2017) use a Gaussian scoring

[1] https://encyclopediaofmath.org.

Work	Learning	Approach	Dataset	Metrics	Math Format
Identifier-Definition Extr.					
Kristianto et al. (2012)	S	CRF with linguistic pattern features	arXiv papers	P, R, F1	MathML
Kristianto et al. (2014a)	S	SVM with linguistic pattern features	arXiv papers	P, R, F1	MathML
Pagael and Schubotz (2014)	R	Gaussian heuristic ranking	Wikipedia articles	P@K, R@K	MathML
Schubotz et al. (2016a)	UNS	Gaussian ranking + K-means clusters	NTCIR-11	P, R, F1	LaTeX
Schubotz et al. (2017)	S	Gaussian rank + pattern matching + SVM	NTCIR-11	P, R, F1	LaTeX
Stathopoulos et al. (2018)	S	Link prediction with BiLSTM	arXiv papers	P, R, F1	MathML
Alexeeva et al. (2020)	R	Odin grammar and open-domain causal IE	MathAlign-Eval	P, R, F1	LaTeX
Jo et al. (2021)	S	BERT fine-tuning	S2ORC	Top1, Top5, MRR	LaTeX
Ferreira et al. (2022)	S	SciBERT fine-tuning with aug. data	arXiv papers	P, R, F1	MathML
van der Goot (2022)	S	Shared encoder + multi-task decoders	Symlink	P, R, F1, F-score	LaTeX
Ping and Chi (2022)	S	BERT fine-tuning with aug. data	Symlink	P, R, F1, F-score	LaTeX
Popovic et al. (2022)	S	SciBERT enc. with entity and relation extr.	Symlink	P, R, F1, F-score	LaTeX
Lee and Na (2022)	S	SciBERT enc. with MRC + tokenizer	Symlink	P, R, F1, F-score	LaTeX
Formula Retrieval					
Kristianto et al. (2014b)	S + R	SVM desc. extr. + leaf-root path search	NTCIR-11	P@5, P@10, MAP	MathML
Kristianto et al. (2016)	S + R	MCAT (2014) + multiple linear regr.	NTCIR-12	P@K	MathML
Zanibbi et al. (2016b)	R	Inverted index rank + MSS rerank search	NTCIR-11	R@K, MRR	MathML
Davila and Zanibbi (2017)	S	Two-stage search for OPT and SLT merged with linear regression	NTCIR-12	P@K, Bpref, nDCG@K	LaTeX
Zhong and Zanibbi (2019)	R	OPT leaf-root path search with K largest subexpressions	NTCIR-12	P@K, Bpref	LaTeX
Mansouri et al. (2019)	UNS	n-gram fastText OPT and SLT embs	NTCIR-12	Bpref	LaTeX/MathML
Peng et al. (2021)	SS	pre-training BERT with tasks related to arXiv math-context pairs and OPTs	NTCIR-12	Bpref	LaTeX
Zhong et al. (2022)	R + S	Approach0 + dense passage retrieval enc.	NTCIR-12	Bpref	LaTeX
Premise Selection					
Ferreira and Freitas (2020b)	S	DGCNN for link prediction	PS-ProofWiki	P, R, F1	LaTeX
Ferreira and Freitas (2021)	S	Self-attention for math and language + BiLSTM with siamese network	PS-ProofWiki	P, R, F1	LaTeX
Coavoux and Cohen (2021)	S	Weighted bipartite matching + self-attention	SPM	MRR, Acc	MathML
Han et al. (2021)	S	LLM fine-tuning (webtext + webmath)	NaturalProofs	R@K, avgP@K, full@K	LaTeX
Welleck et al. (2021a)	S	Fine-tuning BERT with pair/joint param.	NaturalProofs	MAP, R@K, full@K	LaTeX
Dastgheib and Asgari (2022)	UNS	keywords fastText emb. with Jacardian sim	PS-ProofWiki	MAP	LaTeX
Kovriguina et al. (2022)	SS	MathBERT enc. with GPT-3 prompting	PS-ProofWiki	MAP	LaTeX
Kadusabe et al. (2022)	SS	SMPNet with cosine similarity	PS-ProofWiki	MAP	LaTeX
Tran et al. (2022)	SS	RoBERTa with Manhattan similarity	PS-ProofWiki	MAP	LaTeX
MWP Solving					
Liu et al. (2019a)	S	BiLSTM seq enc. + LSTM tree-based dec.	Math23K	Acc	NL
Xie and Sun (2019)	S	GRU encoder + GTS decoder	Math23K	Acc	NL
Li et al. (2020)	S	word-word graph + phrase structure graph Hetero. graph enc. + LSTM tree-based dec.	MAWPS, MATHQA	Acc	NL
Zhang et al. (2020)	S	Word-number graph enc. + Number-comp graph enc. + GTS dec.	MAWPS, Math23K	Acc	NL
Shen and Jin (2020)	S	Seq multi-enc. + tree-based multi-dec.	Math23K	Acc	NL
Kim et al. (2020)	S	ALBERT seq enc. + Transformer seq dec.	ALG514, DRAW-1K, MAWPS	Acc	NL
Qin et al. (2020)	S	Bi-GRU seq enc. + semantically-aligned GTS-based dec.	HMWP, ALG514, Math23K, Dolphin18K	Acc	NL
Cao et al. (2021)	S	GRU-based encoder + DAG-LSTM decoder	DRAW-1K, Math23K	Acc	NL
Lin et al. (2021)	S	Hierarchical GRU seq encoder + GTS decoder	Math23K, MAWPS	Acc	NL
Qin et al. (2021)	S	Bi-GRU enc. + GTS dec. with att. and UET	Math23K, CM17K	Acc	NL
Liang et al. (2021)	S	BERT encoder + GTS decoder	Math23K, APE210K	Acc	NL
Zhang et al. (2022b)	S	word-word + word-num + num-comp graph Heterogeneous graph encoder + GTS decoder	MAWPS, Math23K	Acc	NL
Jie et al. (2022)	S	RoBERTa enc. with bottom-up relation extr.	MAWPS, Math23K, SVAMP, MathQA	Acc	NL
Zhang et al. (2022a)	S	Top-down and bottom-up reasoning with knowledge injection and contrastive learning	MAWPS, Math23K, MathQA	Acc	NL
Informal Theorem Proving					
Wang et al. (2020)	S + UNS	RNNs, LSTMs, transformers	LaTeX, Mizar, TPTP, ProofWiki	BLEU, Perplexity Edit distance	LaTeX
Welleck et al. (2021a)	S	Fine-tuning BERT with pair/joint param.	NaturalProofs	MAP	LaTeX
Welleck et al. (2021b)	SS	BART enc. with denoising pre-training and Fusion-in-Decoder	NaturalProofs	SBleu, Meteor, Edit, P, R, F1	LaTeX
Wu et al. (2022)	S	Fine-tuned LLMs + formal theorem prover	MiniF2F, MATH	Acc	LaTeX/Isabelle
Lewkowycz et al. (2022)	SS/S	Fine-tuned PaLM model	MATH, GSM8K, MMLU-STEM	Acc	LaTeX/NL

Table 1: Summary of different approaches for addressing tasks related to mathematical language processing. The methods are categorized in terms of (i) Learning: Supervised (S), Self-supervised (SS), Unsupervised (UNS), Rule-based (R) (no learning); (iii) Approach; (iv) Dataset; (v) Metrics: MAP (Mean Average Precision), P@K (Precision at K), Perplexity, P (Precision), R (Recall), F1, Acc (Accuracy), BLEU, METEOR, MRR (Mean Reciprocal Rank), Edit (edit distance); (vi) Math format: MathML, LaTeX, natural language (NL), Isabelle formal language. Diagrammatic representations of approaches in identifier-definition extraction (Figure 3), formula retrieval (Figure 4), and MWP solving (Figure 5) can be found in the Appendix.

function (Schubotz et al., 2016b) and pattern matching features (Pagael and Schubotz, 2014) as input to an SVM with a radial basis function (RBF) kernel, to account for non-linear feature characteristics. Alternative classification methods (Kristianto et al., 2012; Stathopoulos et al., 2018) do not use input features derived from non-linear functions, such as the Gaussian scoring function,

and hence use linear kernels. Embedding spaces have been learned in this context for the purpose of *ranking* identifier-definiens pairs through latent semantic analysis at the document level, followed by the application of clustering techniques and methods of relating clusters to namespaces inherited from software engineering (Schubotz et al., 2016a). These cluster-based namespaces are later used for *classification* (Schubotz et al., 2017) rather than ranking, but do not positively impact SVM model performance, despite previous evidence suggesting they resolve coreferences (Duval et al., 2002) such as ''E is energy'' and ''E is expectation value''. Neither clustering nor namespaces have been further explored in this context. More recent work learns context-specific word representations after feeding less specific pre-trained word2vec (Mikolov et al., 2013; Stathopoulos and Teufel, 2016) embeddings to a bidirectional LSTM for classification (Stathopoulos et al., 2018). The most recent work predictably relies on more sophisticated pre-trained BERT embeddings (Devlin et al., 2018) for the language modeling of mathematical notation (Jo et al., 2021). VarSlot (Ferreira et al., 2022) obtains SOTA results on variable typing (Stathopoulos et al., 2018), and demonstrates robustness to variable renaming, by fine-tuning the sentence transformers (Reimers and Gurevych, 2019) SciBERT (Beltagy et al., 2019) encoder on *augmented data*, learning separate representation spaces for variables and mathematical language statements. Four BERT encoder-based approaches (Lee and Na, 2022; Popovic et al., 2022; Ping and Chi, 2022; van der Goot, 2022) were submitted to the Symlink task (Lai et al., 2022), following the trend of knowledge transfer through pre-trained embeddings.

3.2 Formula Retrieval

The task of retrieving similar equations to a query equation, with applications in math-aware search engines (Mansouri et al., 2022a). Guidi and Coen (2016) and Zanibbi and Blostein (2011) emphasize the encoding of formulae and their context for retrieval tasks.

Combining formula tree representations improves retrieval. There are two prevalent types of tree representations of formulae: Symbol Layout Trees (SLTs) and Operator Trees (OPTs), shown in Figure 2.

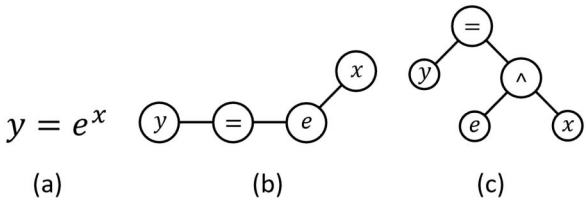

Figure 2: Formula (a) $y = e^x$ with its Symbol Layout Tree (SLT) (b), and Operator Tree (OPT) (c). SLTs represent formula appearance by the spatial arrangements of math symbols, while OPTs define the mathematical operations represented in expressions. For more detail, see Mansouri et al. (2019).

Methods reliant solely on SLTs, such as the early versions of the Tangent retrieval system (Pattaniyil and Zanibbi, 2014; Zanibbi et al., 2015, 2016b), or solely OPTs (Zhong and Zanibbi, 2019; Zhong et al., 2020) tend to return less relevant formulae from queries. OPTs capture formula semantics while SLTs capture visual structure (Mansouri et al., 2019). Effective representation of both formula layout and semantics within a single vector allows a model exploit both representations. Tangent-S (Davila and Zanibbi, 2017) was the first evolution of the Tangent system to outperform the NTCIR-11 (Aizawa et al., 2014) overall best performer, MCAT (Kristianto et al., 2014b; 2016), which encoded path and sibling information from MathML Presentation (SLT-based) and Content (OPT-based). Tangent-S jointly integrated SLTs and OPTs by combining scores for each representation through a simple linear regressor. Later, Tangent-CFT (Mansouri et al., 2019) considered SLTs and OPTs through a fastText (Bojanowski et al., 2017) *n*-gram embedding model using tree tuples. MathBERT (Peng et al., 2021) does *not* explicitly account for SLTs, claiming that LaTeX markup somewhat accounts for SLTs, and therefore encode OPTs. They pretrain the BERT (Devlin et al., 2018) model with targeted objectives each accounting for different aspects of mathematical text. They account for OPTs by concatenating node sequences to formula + context BERT input sequences, and by formulating OPT-based structure-aware pretraining tasks learned in conjunction with masked language modeling (MLM).

Leaf-root path tuples deliver an effective mechanism for embedding relations between symbol pairs. Leaf-root path tuples are now ubiquitous

in formula retrieval (Zanibbi et al., 2015, 2016b; Davila and Zanibbi, 2017; Zhong and Zanibbi, 2019; Mansouri et al., 2019; Zhong et al., 2020) and their use for NTCIR-11/12 retrieval has varied since their conception (Stalnaker and Zanibbi, 2015). Initially (Pattaniyil and Zanibbi, 2014) pair tuples were used within a TF-IDF weighting scheme, then Zanibbi et al. (2015, 2016b) proposed an appearance-based similarity metric using SLTs, *maximum subtree similarity* (MSS). OPT tuples are integrated (Davila and Zanibbi, 2017) later on. Mansouri et al. (2019) treat tree tuples as words, extract *n*-grams, and learn fastText (Bojanowski et al., 2017) formula embeddings. Zhong and Zanibbi (2019) and Zhong et al. (2020) forgo machine learning altogether with an OPT-based heuristic search (Approach0) through a generalization of MSS (Zanibbi et al., 2016b). Leaf-root path tuples effectively map symbol-pair relations and account for formula substructure, but there is dispute on how best to integrate them into existing machine learning or explicit retrieval frameworks.

Purely explicit methods still deliver competitive results. Explicit representation methods are those that rely on prescribed representations (structural relations and associated types) rather than learned implicit relationships. Tangent-CFT (Mansouri et al., 2019) and MathBERT (Peng et al., 2021) are two models to employ learning techniques beyond the level of linear regression. Each model is integrated with Approach0 (Zhong and Zanibbi, 2019) through the linear combination of individual model scores. This respectively forms the TanApp and MathApp baselines in Peng et al. (2021). Approach0 achieves the highest full bpref score of the individual models. While we focus primarily on the NTCIR-12 dataset, recent work (Zhong et al., 2022) evaluates a selection of transformer-based models on both NTCIR-12 and ARQMath-2 (Mansouri et al., 2021b) datasets. They confirm that MathBERT delivers SOTA performance on partial bpref, and Approach0 combined with a fine-tuned dense passage retrieval (DPR) model (Karpukhin et al., 2020) outperforms on full bpref (Approach0 + DPR). Combining explicit similarity-based search (Zhong and Zanibbi, 2019; Meadows and Freitas, 2021) with modern encoders (Khattab and Zaharia, 2020; Karpukhin et al., 2020) delivers leading performance.

3.3 Natural Language Premise Selection

Formal and informal premise selection both involve *the selection of relevant statements for proving a given conjecture* (Irving et al., 2016; Wang et al., 2017a; Ferreira and Freitas, 2020a). The difference lies in the language in which the premises and related proof elements are encoded (either conforming to a logical form or as they appear in mathematical text). Mathematical language as it occurs in papers and textbooks (Wolska and Kruijff-Korbayová, 2004) is not compatible with existing provers without *autoformalization*; a widely acknowledged bottleneck for the construction of formal proof libraries (Irving et al., 2016). Typically, when reasoning over large formal libraries comprising thousands of premises, the performance of ATPs degrades considerably, while for a given proof only a fraction of the premises are required to complete it (Urban et al., 2010; Alama et al., 2014). Theorem proving is essentially a search problem with a combinatorial search space, and the goal of *formal* premise selection is to reduce the space, making theorem proving tractable (Wang et al., 2017a). While formal premises are written in the languages of formal libraries such as Mizar (Rudnicki, 1992), *informal* premises, as seen in ProofWiki,[2] are written in combinations of natural language and LaTeX (Ferreira and Freitas, 2020a; Welleck et al., 2021a). Proposed approaches either rank (Han et al., 2021) or classify (Ferreira and Freitas, 2020b, 2021) candidate premises for a given proof. *Natural language premise selection* was originally formulated as pairwise relevance classification, evaluated with F_1 (Ferreira and Freitas, 2020b, 2021), but has since been evaluated with ranking metrics (Valentino et al., 2022). Alternatively, Welleck et al. (2021a) propose *mathematical reference retrieval* as an analogue of premise selection. The goal is to retrieve the set of references (theorems, lemmas, definitions) that occur in its proof, formulated as a *ranking* problem.

Separate mechanisms for representing mathematics and natural language can improve performance. Regardless of the task variation, most current methods do not fully discriminate the semantics of mathematics and natural language, not specifically accounting for aspects of each modality. Ferreira and Freitas (2020b) extract

[2] https://proofwiki.org/wiki/Main_Page.

a dependency graph representing dual-modality mathematical statements as nodes, and solve a link prediction task (Zhang and Chen, 2018). Recent transformer baselines (Ferreira and Freitas, 2020b; Welleck et al., 2021a; Han et al., 2021; Coavoux and Cohen, 2021), and those at the shared NLPS task (Valentino et al., 2022), also do not differentiate between mathematical elements and natural language (Tran et al., 2022; Kadusabe et al., 2022; Kovriguina et al., 2022). STAR (Ferreira and Freitas, 2021) purposefully separates the two modalities, encoding distinct representations through self-attention. Explicit disentanglement of the modalities forces STAR to exploit relationships between natural language and mathematics, through the BiLSTM layer. Neuroscience research suggests the brain handles mathematics separately to language (Butterworth, 2002; Amalric and Dehaene, 2016; Kulasingham et al., 2021).

3.4 Math Word Problems

Solving math word problems dates back to the dawn of artificial intelligence research (Feigenbaum and Feldman, 1963; Bobrow, 1964; Charniak, 1969). It can be defined as the task of translating a problem description paragraph into a set of equations to be solved (Li et al., 2020). We focus on trends in the task since 2019, as a detailed survey (Zhang et al., 2019) captures prior work.

Use of dependency graphs is instrumental to support inference. In graph-based approaches to solving MWPs, embeddings of words, numbers, or relationship graph nodes are learned through *graph encoders*, which feed information through to tree (or sequence) decoders. Embeddings are decoded into expression trees which determine the problem solution. Li et al. (2020) learn the mapping between a heterogeneous graph representing the input problem, and an output tree. The graph is constructed from word nodes with relationship nodes of a parsing tree. This is either a dependency parse tree or constituency tree. Zhang et al. (2020) represent *two* separate graphs: a *quantity cell graph* associating descriptive words with problem quantities, and a *quantity comparison graph* which retains numerical qualities of the quantity, and leverages heuristics to represent relationships between quantities such that solution expressions reflect a more realistic arithmetic order. Shen and Jin (2020) also extract *two* graphs: a dependency parse tree and numerical comparison graph. Zhang et al. (2022b) construct a heterogeneous graph from three subgraphs: a *word-word graph* containing syntactic and semantic relationships between words, a *number-word graph*, and a *number comparison graph*. Although other important differences exist (such as decoder choice), it seems models benefit from relating linguistic aspects of problem text through separate graphs.

Multi-encoders and multi-decoders improve performance by combining complementary representations. Another impactful decision is the choice of encoder/decoder, and whether to consider alternative representations of a problem. To highlight this, we consider the following comparison. Shen and Jin (2020) and Zhang et al. (2020) each extract two graphs from the problem text. One is a number comparison graph, and the other relates word-word pairs (Shen and Jin, 2020) or word-number pairs (Zhang et al., 2020). They both encode *two* graphs rather than one heterogeneous graph (Li et al., 2020; Zhang et al., 2022b). They both use a similar tree-based decoder (Xie and Sun, 2019). A key difference is that Shen and Jin (2020) include *an additional sequence-based encoder and decoder*. The sequence-based encoder first obtains a textual representation of the input paragraph, then the graph-based encoder integrates the two encoded graphs. Then tree-based and sequence-based decoders generate *different equation expressions* for the problem with an additional mechanism for optimizing solution expression selection. In their own work, Shen and Jin (2020) demonstrate the impact of multi-encoders/decoders over each encoder/decoder option individually through ablation. Zhang et al. (2022a) similarly combine top-down and bottom-up reasoning to achieve leading results.

Goal-driven decompositional tree-based decoders are a significant component in the state-of-the-art. Introduced in Xie and Sun (2019), this class of decoder is considered by most of the discussed approaches, and includes non-graph-based models (Qin et al., 2021; Liang et al., 2021). In GTS, goal vectors guide construction of expression subtrees (from token node embeddings) in a recursive manner, until a solution expression tree

is generated. Proposed models do expand on the GTS-based decoder through the inclusion of semantically aligned universal expression trees (Qin et al., 2020, 2021), though this adaptation is not as widely used. Some state-of-the-art (Liang et al., 2021; Zhang et al., 2022b) models follow the GTS decoder closely.

Language models that transfer knowledge learned from auxiliary tasks rival models based on explicit graph representation of problem text. As an alternative to encoding explicit relations through graphs, other work (Kim et al., 2020; Qin et al., 2021; Liang et al., 2021) relies on pre-trained transformer-based models, and those which incorporate auxiliary tasks assumed relevant for solving MWPs, to learn such relations latently. However, it seems the case that auxiliary tasks alone do not deliver competitive performance (Qin et al., 2020) without the extensive pre-training efforts with large corpora, as we see with BERT-based transformer models. These use either both the ALBERT (Lan et al., 2019) encoder and decoder (Kim et al., 2020), or BERT-based encoder with goal-driven tree-based decoder (Liang et al., 2021). More recent work (Cao et al., 2021; Jie et al., 2022; Zhang et al., 2022a) involves *iterative relation extraction* frameworks for predicting mathematical relations between numerical tokens.

3.5 Informal Theorem Proving

Formal automated theorem proving in logic is among the most abstract forms of reasoning materialised in the AI space. There are two major bottlenecks (Irving et al., 2016) that formal methods must overcome: (1) translating informal mathematical text into formal language (*autoformalization*), and (2) a lack of strong automated reasoning methods to fill in the gaps in already formalized human-written proofs. Informal methods either tackle autoformalization directly (Wang et al., 2020; Wu et al., 2022), or circumvent it through language modeling-based proof generation (Welleck et al., 2021a,b), trading formal rigor and inference control for flexibility. Transformer-based models have been proposed for mathematical reasoning (Polu and Sutskever, 2020; Rabe et al., 2020; Wu et al., 2021). Converting *informal* mathematical text into forms which are interpretable by computers (Kaliszyk et al., 2015a,b; Szegedy, 2020; Wang and Deng, 2020; Meadows and Freitas, 2021) can strategically impact the dialogue between knowledge expressed in natural text, and a large spectrum of solvers.

Autoformalization could be addressed through approximate translation and exploration rather than direct machine translation. A long-studied and challenging endeavour (Zinn, 1999, 2003), autoformalization involves converting informal mathematical text into language interpretable by theorem provers (Kaliszyk et al., 2015b; Wang et al., 2020; Szegedy, 2020). Kaliszyk et al. (2015b) propose statistical learning methods for parsing ambiguous formulae over the Flyspeck formal mathematical corpus (Hales, 2006). Using machine translation models (Luong et al., 2017; Lample et al., 2018; Lample and Conneau, 2019), Wang et al. (2020) explore dataset translation experiments between LaTeX code extracted from ProofWiki, and formal libraries Mizar (Rudnicki, 1992) and TPTP (Sutcliffe and Suttner, 1998). The supervised RNN-based neural machine translation model (Luong et al., 2017) outperforms the transformer-based (Lample et al., 2018) and MLM pre-trained transformer-based (Lample and Conneau, 2019) models, with the performance boost stemming from its use of alignment data. Szegedy (2020) advises against such direct translation efforts, instead proposing a combination of exploration and approximate translation through predicting formula embeddings. In seq2seq models, embeddings are typically granular, encoding word-level or symbol-level (Jo et al., 2021) tokens. The method consists of learning mappings from natural language input to premise statements nearby the desired statement in the embedding space, traversing the space between statements using a suitable prover (Bansal et al., 2019). Guided mathematical exploration for real-world proofs is still an unaddressed problem and does not scale well with step-distance between current and desired conjecture. Wu et al. (2022) directly autoformalize small competition problems to Isabelle statements using language models. Similar to previous indication (Szegedy, 2020), they also autoformalize statements as targets for proof search with a neural theorem prover.

The need for developing robust interactive natural language theorem provers. We discuss the closest equivalent to formal theorem proving in an informal setting. Welleck et al. (2021a) propose

a *mathematical reference generation* task. Given a mathematical claim, the order and number of references within a proof are predicted. A reference is a theorem, definition, or a page that is linked to within the contents of a statement or proof. Each theorem $\mathbf{x}$ has a proof containing a sequence of references $\mathbf{y} = (\mathbf{r_1}, \ldots, \mathbf{r}_{|\mathbf{y}|})$, for references $\mathbf{r_m} \in \mathcal{R}$. Where the *retrieval* task assigns a score to each reference in $\mathcal{R}$, the *generation* task produces a variable length of sequence of references $(\hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{|\mathbf{y}|})$ with the goal of matching $\mathbf{y}$, for which a BERT-based model is employed and fine-tuned on various data sources. Welleck et al. (2021b) expand on their proof generation work, proposing two related tasks: *next-step suggestion*, where a step from a proof $\mathbf{y}$ (as described above) is defined as a sequence of tokens to be generated, given the previous steps and $\mathbf{x}$; and *full-proof generation* which extends this to generate the full proof. They employ BART (Lewis et al., 2019), an encoder-decoder model pre-trained with *denoising* tasks, and augment the model with reference knowledge using Fusion-in-Decoder (Izacard and Grave, 2020). The intermediate denoising training and knowledge-grounding improve model performance by producing better representations of (denoised) references for deployment at generation time, and by encoding reference-augmented inputs. Minerva (Lewkowycz et al., 2022) is a language model capable of producing step-wise reasoning with mathematical language (LaTeX). They fine-tune a PaLM decoder-only model (Chowdhery et al., 2022) on webpages containing MathJax formatted expressions, and evaluate on school-level math problems (Hendrycks et al., 2021; Cobbe et al., 2021), a STEM subset of problems (Hendrycks et al., 2020) of varying difficulty, undergraduate-level STEM problems, and the National Math Exam in Poland. They evaluate for *generalization capabilities* by generating 20 alternative evaluation problems, perturbing problem wording and numerical values in the MATH (Hendrycks et al., 2021) dataset, and compare accuracy before and after the change. While they suggest ''minimal memorization'', the numerical intervention comparison does less to support this claim.

4 Datasets

Various datasets have been proposed for tasks related to *identifier-definition extraction* and variable typing (Schubotz et al., 2016a; Alexeeva et al., 2020; Stathopoulos et al., 2018; Jo et al., 2021), with limited adoption. The Symlink shared task (Lai et al., 2022) is an emerging solution, with training data, annotations of 102 papers, and high inter-annotator agreement. *Formula retrieval* data exists through NTCIR-12 (Zanibbi et al., 2016a), which has been expanded in the most recent ARQMath task (Mansouri et al., 2022b), removing formula duplicates and balancing query complexity. *Premise selection* datasets include PS-ProofWiki (Ferreira and Freitas, 2020a), used in the NLPS shared task (Valentino et al., 2022), and NaturalProofs (Welleck et al., 2021a). The latter is more inclusive, comprising ProofWiki, text books, and other sources. Modern consensus *MWP* datasets include (easy) MAWPS (Koncel-Kedziorski et al., 2016), (medium) Math23K (Wang et al., 2017b), and (hard) MathQA (Amini et al., 2019), comprising both Chinese and English problems. GSM8K (Cobbe et al., 2021) claims to resolve diversity, quality, and language (Huang et al., 2016) issues from previous datasets, involves step-wise reasoning and natural language solutions, with balanced difficulty. MATH (Hendrycks et al., 2021) is larger and more difficult than GSM8K. *Informal theorem proving* data includes NaturalProofs (Welleck et al., 2021a), and some MWP datasets involving step-wise reasoning with mathematical language, such as MATH and GSM8K. However, there is no consensus data for autoformalization or theorem proving from mathematical language input involving sequence learning. ProofNet (Azerbayev et al., 2022) aims to remedy this, by providing 297 theorem statements expressed in both natural and formal (Moura et al., 2015) language, at undergraduate difficulty. Some are accompanied by informal proofs. MiniF2F (Zheng et al., 2021) is a neural theorem proving benchmark of Olympiad-level problems written in many formal languages. Lila (Mishra et al., 2022) provides data for 23 math reasoning tasks. Key datasets information is described in Table 2.

Data Scarcity. Some datasets, such as MATH and the Auxiliary Mathematics Problems and Solutions (AMPS) (Hendrycks et al., 2021) datasets, include detailed workings at high school to undergraduate level difficulty. If we aim to use models to produce new mathematics, equivalent datasets composed of the research workings

Name	Tasks	Size
Symlink	Identifier-Def Extr.	31K entities, 20K relations
ARQMath-2 Task 2	Formula Retrieval	100 queries, 28M formulae
NTCIR-12	Formula Retrieval	40 formula queries, 590K formulae
PS-ProofWiki	Premise Selection	14K theorems, 5K definitions 300 lemmas, 292 corollaries
NaturalProofs	Premise Selection Proof Generation	32K theorems/proofs, 14K definitions 2K corollories + axioms
Math23K	Math Word Problems	23K problems
MAWPS	Math Word Problems	3K problems
MathQA	Math Word Problems	37K problems
GSM8K	Math Word Problems	8K problems
MATH	Math Word Problems Proof Generation	13K hard problems
ProofNet	Proof Generation	297 theorems/proofs

Table 2: Key datasets for the representative tasks.

of actual mathematicians would be invaluable. Meadows and Freitas (2021) attempt to tackle this problem for a single research paper in a very limited setting.

5 Discussion

State-of-the-art. In *identifier-definition extraction*, leading performance is obtained on Symlink by Lee and Na (2022), using a SciBERT encoder and MRC-based model (Li et al., 2019). Importantly, rather than the BERT tokenizer, they use a *rule-based symbol tokenizer*, evidencing the benefits of discerning natural language from math elements. VarSlot (Ferreira et al., 2022) leads in variable typing, and echoes the importance of such discrimination (see Section 3.2). In *formula retrieval*, SOTA methods generally include linear combinations of scores obtained from symbolic and neural models. On NTCIR-12, Zhong et al. (2022) show that MathBERT leads on partial bpref, and Approach0 + DPR leads on full bpref (see Section 3.2). Approach0 + ColBERT (Khattab and Zaharia, 2020) leads on ARQMath-2 (Mansouri et al., 2021b). This work reinforces the importance of including formula structure across multiple tasks. In *premise selection*, leading results are obtained on the shared NLPS task by a fine-tuned RoBERTa-large encoder (Liu et al., 2019b), computing similarity scores between statements with Manhattan distance (Tran et al., 2022). However, none of the competing models discern mathematical elements from natural language, or include formula structure. In *MWP solving*, the multi-view model (Zhang et al., 2022a) achieves state-of-the-art results on Math23K, MAWPS, and MathQA. Minerva, and the Diverse approach (Li et al., 2022) based on OpenAI code-davinci-002, lead on MATH. Minerva also beats the national 57% average by 8% on the Polish national math exam. In *informal theorem proving*, we discuss autoformalization and theorem proving from mathematical language. In the former, code-davinci-002 leads on ProofNet. In the latter, a BART-based model leads on NaturalProofs, and Codex (Chen et al., 2021) fine-tuned on autoformalized theorems (Wu et al., 2022), leads on MiniF2F. These later methods, particularly those that score highly on MATH, largely consist of fine-tuning generative LLMs also without distinctly considering mathematical content or structure.

Separate Representations for Math and Natural Language. Many models do not benefit from processing each modality separately. The leading model on Symlink uses a special tokenizer to extract math symbols from scientific

documents (Lee and Na, 2022). VarSlot improves variable typing by learning representation spaces for variables and mathematical language statements (Ferreira et al., 2022). STAR (Ferreira and Freitas, 2021) improves on a self-attention baseline encoding combined math/language statements, by separately encoding math and language with the same encoder. MathBERT learns embeddings from tree and latex representations of formulae, and natural language (Peng et al., 2021). The Approach0 + [encoder] models linearly combine scores from entirely different methods; one designed for formulae, and one for language (Zhong et al., 2022). Multi-view learns an embedding each for words, quantities, and operations (Zhang et al., 2022a). All of the above are state-of-the-art and show advantage over baselines that do not invoke separate mechanisms. Despite this evidence, methods related to *informal theorem proving* and *premise selection*, such as Minerva, IJS (Tran et al., 2022), and others, do not discriminate math from language. This is likely true for other subfields of MLP.

Math as Trees. Many approaches do not incorporate formula structure. For problems involving multi-variate mathematical terms, obvious choices for this are OPTs and SLTs (Figure 2). For example, Approach0 considers formula OPTs, *without learning*, to achieve competitive results. Inclusion of OPTs during BERT training has been shown to improve performance over BERT in formula retrieval, formula headline generation, and formula topic classification (Peng et al., 2021), and is also used in math question answering (Mansouri et al., 2021a).

Combining Complementary Representations from the Same Input. Combined use of OPTs and SLTs of the same formula has been suggested to improve formula retrieval performance (Davila and Zanibbi, 2017; Mansouri et al., 2019; Mansouri et al., 2021a). This extends to dual-modality mathematical language input. Shen and Jin (2020) obtain sequence and graph encodings of MWPs, and use sequence and tree-based decoders in unison, with an ablation describing advantage over single encoder representations. The leading MWP solver (Zhang et al., 2022a) generates two independent solution expression embeddings, by top-down decomposition (Xie and Sun, 2019) and bottom-up construction, which are projected into the same latent space.

Conclusion. Delivering mathematical reasoning over discourse requires close integration between step-wise inference control over localised explicit representations (symbolic perspective), and distributed representations to approximate and cope with incomplete knowledge (neural perspective). The current spectrum of mathematical language processing techniques elicits the key components, representational choices and tasks which are central to the conceptualisation of mathematical inference. Integrating the best-performing representational choices across different subtasks, such as distinct mechanisms for processing natural language and formulae, learning complementary representations of mathematical problem text, and incorporating formula structure, represents a short-term opportunity to develop mathematically robust models capable of more coherent argumentation, reasoning, and retrieval.

Acknowledgments

This work was partially funded by the Swiss National Science Foundation (SNSF) project NeuMath (200021_204617).

References

Akiko Aizawa, Michael Kohlhase, Iadh Ounis, and Moritz Schubotz. 2014. Ntcir-11 math-2 task overview. In *NTCIR*, volume 11, pages 88–98.

Alan Akbik, Xinyu Guan, and Yunyao Li. 2016. Multilingual aliasing for auto-generating proposition banks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3466–3474.

Jesse Alama, Tom Heskes, Daniel Kühlwein, Evgeni Tsivtsivadze, and Josef Urban. 2014. Premise selection for mathematics by corpus analysis and kernel methods. *Journal of Automated Reasoning*, 52(2):191–213. https://doi.org/10.1007/s10817-013-9286-5

Maria Alexeeva, Rebecca Sharp, Marco A. Valenzuela-Escárcega, Jennifer Kadowaki, Adarsh Pyarelal, and Clayton Morrison. 2020. Mathalign: Linking formula identifiers to their contextual natural language descriptions. In

Proceedings of The 12th Language Resources and Evaluation Conference, pages 2204–2212.

Marie Amalric and Stanislas Dehaene. 2016. Origins of the brain networks for advanced mathematics in expert mathematicians. *Proceedings of the National Academy of Sciences*, 113(18):4909–4917. https://doi.org/10.1073/pnas.1603205113, PubMed: 27071124

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*.

Takuto Asakura, Yusuke Miyao, Akiko Aizawa, and Michael Kohlhase. 2021. Miogatto: A math identifier-oriented grounding annotation tool. Technical report, EasyChair.

Zhangir Azerbayev, Bartosz Piotrowski, and Jeremy Avigad. 2022. Proofnet: A benchmark for autoformalizing and formally proving undergraduate-level mathematics problems.

Kshitij Bansal, Sarah M. Loos, Markus N. Rabe, Christian Szegedy, and Stewart Wilcox. 2019. Holist: An environment for machine learning of higher-order theorem proving (extended version). *arXiv preprint arXiv:1904.03241*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*. https://doi.org/10.18653/v1/D19-1371

Daniel G. Bobrow. 1964. Natural language input for a computer problem solving system.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146. https://doi.org/10.1162/tacl_a_00051

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Brian Butterworth. 2002. Mathematics and the brain. *Opening address to the Mathematical Association, Reading*.

Yixuan Cao, Feng Hong, Hongwei Li, and Ping Luo. 2021. A bottom-up dag structure extraction model for math word problems. In *Thirty-Fifth AAAI Conference on Artificial 2021*, pages 39–46. https://doi.org/10.1609/aaai.v35i1.16075

Eugene Charniak. 1969. Computer solution of calculus word problems. In *Proceedings of the 1st International Joint Conference on Artificial Intelligence*, pages 303–316.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben

Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. *arXiv preprint arXiv: 2002.05867*. https://doi.org/10.24963/ijcai.2020/537

Maximin Coavoux and Shay B. Cohen. 2021. Learning to match mathematical statements with proofs. *arXiv preprint arXiv:2102.02110*.

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*.

Doratossadat Dastgheib and Ehsaneddin Asgari. 2022. Keyword-based natural language premise selection for an automatic mathematical statement proving. In *Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing*, pages 124–126.

Kenny Davila and Richard Zanibbi. 2017. Layout and semantics: Combining representations for mathematical formula search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1165–1168. https://doi.org/10.1145/3077136.3080748

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv: 1810.04805*.

Zhendong Dong and Qiang Dong. 2003. Hownet-a hybrid language and knowledge resource. In *International Conference on Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003*, pages 820–824. IEEE. https://doi.org/10.1109/NLPKE.2003.1276017

Erik Duval, Wayne Hodgins, Stuart Sutton, and Stuart L. Weibel. 2002. Metadata principles and practicalities. *D-lib Magazine*, 8(4):1–10. https://doi.org/10.1045/april2002-weibel

Edward A. Feigenbaum and Julian Feldman, eds. 1963. *Computers and Thought*. New York McGraw-Hill.

Weijie Feng, Binbin Liu, Dongpeng Xu, Qilong Zheng, and Yun Xu. 2021. Graphmr: Graph neural network for mathematical reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3395–3404. https://doi.org/10.18653/v1/2021.emnlp-main.273

Deborah Ferreira and Andre Freitas. 2020a. Natural language premise selection: Finding supporting statements for mathematical text. *arXiv preprint arXiv:2004.14959*.

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*, pages 7365–7374. https://doi.org/10.18653/v1/2020.acl-main.657

Deborah Ferreira and André Freitas. 2021. Star: Cross-modal [sta] tement [r] epresentation for selecting relevant mathematical premises. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3234–3243. https://doi.org/10.18653/v1/2021.eacl-main.282

Deborah Ferreira, Mokanarangan Thayaparan, Marco Valentino, Julia Rozanova, and Andre Freitas. 2022. To be or not to be an integer? Encoding variables for mathematical text. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 938–948,

Dublin, Ireland. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-acl.76

Rob van der Goot. 2022. Machamp at semeval-2022 tasks 2, 3, 4, 6, 10, 11, and 12: Multi-task multi-lingual learning for a pre-selected set of semantic datasets. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022). Association for Computational Linguistics*. https://doi.org/10.18653/v1/2022.semeval-1.233

Adrian Groza and Cristian Nitu. 2022. Question answering over logic puzzles using theorem proving. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pages 871–874. https://doi.org/10.1145/3477314.3507177

Ferruccio Guidi and Claudio Sacerdoti Coen. 2016. A survey on retrieval of mathematical knowledge. *Mathematics in Computer Science*, 10(4):409–427. https://doi.org/10.1007/s11786-016-0274-0

Thomas C. Hales. 2006. Introduction to the flyspeck project. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Jesse Michael Han, Tao Xu, Stanislas Polu, Arvind Neelakantan, and Alec Radford. 2021. Contrastive finetuning of generative language models for informal premise selection.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Tran Hong-Minh and Dan Smith. 2008. Word similarity in wordnet. In *Modeling, Simulation and Optimization of Complex Processes*, pages 293–302. Springer. https://doi.org/10.1007/978-3-540-79409-7_19

Mark Hopkins, Ronan Le Bras, Cristian Petrescu-Prahova, Gabriel Stanovsky, Hannaneh Hajishirzi, and Rik Koncel-Kedziorski. 2019. Semeval-2019 task 10: Math question answering. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 893–899. https://doi.org/10.18653/v1/S19-2153

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 (Volume 1: Long Papers)*, pages 887–896. https://doi.org/10.18653/v1/P16-1084

Geoffrey Irving, Christian Szegedy, Alexander A. Alemi, Niklas Eén, François Chollet, and Josef Urban. 2016. Deepmath-deep sequence models for premise selection. In *Advances in Neural Information Processing Systems*, pages 2235–2243.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*. https://doi.org/10.18653/v1/2021.eacl-main.74

Zhanming Jie, Jierui Li, and Wei Lu. 2022. Learning to reason deductively: Math word problem solving as complex relation extraction. *arXiv preprint arXiv:2203.10316*.

Hwiyeol Jo, Dongyeop Kang, Andrew Head, and Marti A. Hearst. 2021. Modeling mathematical notation semantics in academic papers. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3102–3115.

Provia Kadusabe, Haseeb Younis, Rosane Minghim, Evangelos Milios, Ahmed Zahran, et al. 2022. Snlp at textgraphs 2022 shared task: Unsupervised natural language premise selection in mathematical texts using sentence-mpnet. In *Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing*, pages 119–123.

Cezary Kaliszyk, Josef Urban, Umair Siddique, Sanaz Khan-Afshar, Cvetan Dunchev, and Sofiene Tahar. 2015a. Formalizing physics: Automation, presentation and foundation issues. In *International Conference on Intelligent Computer Mathematics*, pages 288–295.

Springer. https://doi.org/10.1007/978-3-319-20615-8_19

Cezary Kaliszyk, Josef Urban, and Jiří Vyskočil. 2015b. Learning to parse on aligned corpora (rough diamond). In *International Conference on Interactive Theorem Proving*, pages 227–233. Springer. https://doi.org/10.1007/978-3-319-22102-1_15

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*. https://doi.org/10.18653/v1/2020.emnlp-main.550

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–48. https://doi.org/10.1145/3397271.3401075

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. https://doi.org/10.18653/v1/2020.emnlp-main.308

Rik Koncel-Kedziorski, 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*, pages 1152–1157. https://doi.org/10.18653/v1/N16-1136

Liubov Kovriguina, Roman Teucher, and Robert Wardenga. 2022. Textgraphs-16 natural language premise selection task: Zero-shot premise selection with prompting generative language models. In *Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing*, pages 127–132.

Giovanni Yoko Kristianto, Akiko Aizawa, et al. 2014a. Extracting textual descriptions of mathematical expressions in scientific papers. *D-Lib Magazine*, 20(11):9. https://doi.org/10.1045/november14-kristianto

Giovanni Yoko Kristianto, Minh-Quoc Nghiem, Yuichiroh Matsubayashi, and Akiko Aizawa. 2012. Extracting definitions of mathematical expressions in scientific papers. In *Proceedings of the 26th Annual Conference of JSAI*, pages 1–7.

Giovanni Yoko Kristianto, Goran Topic, and Akiko Aizawa. 2016. Mcat math retrieval system for ntcir-12 mathir task. In *NTCIR*.

Giovanni Yoko Kristianto, Goran Topic, Florence Ho, and Akiko Aizawa. 2014b. The mcat math retrieval system for ntcir-11 math track. In *NTCIR*.

Joshua P. Kulasingham, Neha H. Joshi, Mohsen Rezaeizadeh, and Jonathan Z. Simon. 2021. Cortical processing of arithmetic and simple sentences in an auditory attention task. *Journal of Neuroscience*, 41(38):8023–8039. https://doi.org/10.1523/JNEUROSCI.0269-21.2021, PubMed: 34400518

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 (Volume 1: Long Papers)*, pages 271–281. https://doi.org/10.3115/v1/P14-1026

Viet Lai, Amir Pouran Ben Veyseh, Franck Dernoncourt, and Thien Nguyen. 2022. Semeval 2022 task 12: Symlink-linking mathematical symbols to their descriptions. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1671–1678. https://doi.org/10.18653/v1/2022.semeval-1.230

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. *arXiv preprint arXiv:1804.07755*. https://doi.org/10.18653/v1/D18-1549

Andrew S. Lan, Divyanshu Vats, Andrew E. Waters, and Richard G. Baraniuk. 2015.

Mathematical language processing: Automatic grading and feedback for open response mathematical questions. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pages 167–176. https://doi.org/10.1145/2724660.2724664

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*.

Sung-Min Lee and Seung-Hoon Na. 2022. Jbnu-cclab at semeval-2022 task 12: Fusing maximum entity information for linking mathematical symbols to their descriptions.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*. https://doi.org/10.18653/v1/2020.acl-main.703

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. 2022. Solving quantitative reasoning problems with language models. *arXiv preprint arXiv: 2206.14858*.

Shucheng Li, Lingfei Wu, Shiwei Feng, Fangli Xu, Fengyuan Xu, and Sheng Zhong. 2020. Graph-to-tree neural networks for learning structured input-output translation with applications to semantic parsing and math word problem. *arXiv preprint arXiv:2004.13781*.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2019. A unified mrc framework for named entity recognition. *arXiv preprint arXiv:1910.11476*.

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

Zhenwen Liang, Jipeng Zhang, Jie Shao, and Xiangliang Zhang. 2021. Mwp-bert: A strong baseline for math word problems. *arXiv preprint arXiv:2107.13435*.

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 *Thirty-Fifth AAAI Conference on Artificial 2021*, pages 4232–4240. https://doi.org/10.1609/aaai.v35i5.16547

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. *arXiv preprint arXiv:1907.11692*.

Minh-Thang Luong, Eugene Brevdo, and Rui Zhao. 2017. Neural machine translation (seq2seq) tutorial.

Behrooz Mansouri, Anurag Agarwal, Douglas W. Oard, and Richard Zanibbi. 2022a. Advancing math-aware search: The arqmath-3 lab at clef 2022. In *Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part II*, pages 408–415. Springer. https://doi.org/10.1007/978-3-030-99739-7_51

Behrooz Mansouri, Vít Novotnỳ, Anurag Agarwal, Douglas W. Oard, and Richard Zanibbi. 2022b. Overview of arqmath-3 (2022): Third clef lab on answer retrieval for questions on math (working notes version). *Working Notes of CLEF*. https://doi.org/10.1007/978-3-031-13643-6_20

Behrooz Mansouri, Douglas W. Oard, and Richard Zanibbi. 2021a. Dprl systems in the clef 2022 arqmath lab: Introducing mathamr for math-aware search. *Proceedings of the Working Notes of CLEF 2022*, pages 5–8.

Behrooz Mansouri, Shaurya Rohatgi, Douglas W. Oard, Jian Wu, C. Lee Giles, and Richard Zanibbi. 2019. Tangent-cft: An embedding model for mathematical formulas. In *Proceedings of the 2019 ACM SIGIR International*

Conference on Theory of Information Retrieval, pages 11–18. https://doi.org/10.1145/3341981.3344235

Behrooz Mansouri, Richard Zanibbi, Douglas W. Oard, and Anurag Agarwal. 2021b. Overview of arqmath-2 (2021): second clef lab on answer retrieval for questions on math. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 215–238. Springer. https://doi.org/10.1007/978-3-030-85251-1_17

Jordan Meadows and André Freitas. 2021. Similarity-based equational inference in physics. *Physical Review Research*, 3(4):L042010. https://doi.org/10.1103/PhysRevResearch.3.L042010

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Swaroop Mishra, Matthew Finlayson, Pan Lu, Leonard Tang, Sean Welleck, Chitta Baral, Tanmay Rajpurohit, Oyvind Tafjord, Ashish Sabharwal, Peter Clark, et al. 2022. Lila: A unified benchmark for mathematical reasoning. *arXiv preprint arXiv:2210.17517*.

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. https://doi.org/10.1007/978-3-319-21401-6_26

Robert Pagael and Moritz Schubotz. 2014. Mathematical language processing project. *arXiv preprint arXiv:1407.0167*.

Nidhin Pattaniyil and Richard Zanibbi. 2014. Combining tf-idf text retrieval with an inverted index over symbol pairs in math expressions: The tangent math search engine at ntcir 2014. In *NTCIR*.

Shuai Peng, Ke Yuan, Liangcai Gao, and Zhi Tang. 2021. Mathbert: A pre-trained model for mathematical formula understanding. *arXiv preprint arXiv:2105.00377*.

Annie Ping and Ethan Chi. 2022. Team an (l) p at semeval-2022 task 12: Building a lightweight symbol recognition system.

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

Nicholas Popovic, Walter Laurito, and Michael Färber. 2022. Aifb-webscience at semeval-2022 task 12: Relation extraction first–using relation extraction to identify entities. *arXiv preprint arXiv:2203.05325*. https://doi.org/10.18653/v1/2022.semeval-1.232

Jinghui Qin, Xiaodan Liang, Yining Hong, Jianheng Tang, and Liang Lin. 2021. Neural-symbolic solver for math word problems with auxiliary tasks. *arXiv preprint arXiv:2107.01431*.

Jinghui Qin, Lihui Lin, Xiaodan Liang, Rumin Zhang, and Liang Lin. 2020. Semantically-aligned universal tree-structured solver for math word problems. *arXiv preprint arXiv:2010.06823*.

Markus N. Rabe, Dennis Lee, Kshitij Bansal, and Christian Szegedy. 2020. Mathematical reasoning via self-supervised skip-tree training. *arXiv preprint arXiv:2006.04757*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*. https://doi.org/10.18653/v1/D19-1410

Subhro Roy and Dan Roth. 2016. Solving general arithmetic word problems. *arXiv preprint arXiv:1608.01413*.

Piotr Rudnicki. 1992. An overview of the mizar project. In *Proceedings of the 1992 Workshop on Types for Proofs and Programs*, pages 311–330.

Moritz Schubotz, Alexey Grigorev, Marcus Leich, Howard S. Cohl, Norman Meuschke, Bela Gipp, Abdou S. Youssef, and Volker Markl. 2016a. Semantification of identifiers in mathematics for better math information retrieval. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 135–144. https://doi.org/10.1145/2911451.2911503

Moritz Schubotz, Leonard Krämer, Norman Meuschke, Felix Hamborg, and Bela Gipp. 2017. Evaluating and improving the extraction

of mathematical identifier definitions. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 82–94. Springer. https://doi.org/10.1007/978-3-319-65813-1_7

Moritz Schubotz, David Veenhuis, and Howard S. Cohl. 2016b. Getting the units right. In *FM4M/MathUI/ThEdu/DP/WIP@ CIKM*, pages 146–156.

Moritz Schubotz, Abdou Youssef, Volker Markl, and Howard S. Cohl. 2015. Challenges of mathematical information retrievalin the ntcir-11 math wikipedia task. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 951–954. https://doi.org/10.1145/2766462.2767787

Rebecca Sharp, Adarsh Pyarelal, Benjamin Gyori, Keith Alcock, Egoitz Laparra, Marco A. Valenzuela-Escárcega, Ajay Nagesh, Vikas Yadav, John Bachman, Zheng Tang, Heather Lent, Fan Luo, Mithun Paul, Steven Bethard, Kobus Barnard, C. Morrison, and M. Surdeanu. 2019. Eidos, indra, & delphi: From free text to executable causal models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. https://doi.org/10.18653/v1/N19-4008

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*, pages 2924–2934. https://doi.org/10.18653/v1/2020.coling-main.262

David Stalnaker and Richard Zanibbi. 2015. Math expression retrieval using an inverted index over symbol pairs. In *Document Recognition and Retrieval XXII*, volume 9402, pages 34–45. SPIE. https://doi.org/10.1117/12.2074084

Yiannos Stathopoulos, Simon Baker, Marek Rei, and Simone Teufel. 2018. Variable typing: Assigning meaning to variables in mathematical text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 303–312. https://doi.org/10.18653/v1/N18-1028

Yiannos Stathopoulos and Simone Teufel. 2015. Retrieval of research-level mathematical information needs: A test collection and technical terminology experiment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 334–340. https://doi.org/10.3115/v1/P15-2055

Yiannos Stathopoulos and Simone Teufel. 2016. Mathematical information retrieval based on type embeddings and query expansion. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2344–2355.

Geoff Sutcliffe and Christian Suttner. 1998. The tptp problem library. *Journal of Automated Reasoning*, 21(2):177–203. https://doi.org/10.1023/A:1005806324129

Christian Szegedy. 2020. A promising path towards autoformalization and general artificial intelligence. In *International Conference on Intelligent Computer Mathematics*, pages 3–20. Springer. https://doi.org/10.1007/978-3-030-53518-6_1

Thi Hong Hanh Tran, Matej Martinc, Antoine Doucet, and Senja Pollak. 2022. Ijs at textgraphs-16 natural language premise selection task: Will contextual information improve natural language premise selection? In *Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing*, pages 114–118.

Josef Urban, Krystof Hoder, and Andrei Voronkov. 2010. Evaluation of automated theorem proving on the mizar mathematical library. In *International Congress on Mathematical Software*, pages 155–166. Springer. https://doi.org/10.1007/978-3-642-15582-6_30

Marco Valentino, Deborah Ferreira, Mokanarangan Thayaparan, André Freitas, and Dmitry Ustalov. 2022. TextGraphs 2022 shared task on natural language premise selection. In *Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing*, pages 105–113, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Marco A. Valenzuela-Escárcega, Gus Hahn-Powell, and Mihai Surdeanu. 2016. Odin's runes: A rule language for information extraction. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 322–329.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv: 1706.03762*.

Mingzhe Wang and Jia Deng. 2020. Learning to prove theorems by learning to generate theorems. *arXiv preprint arXiv:2002.07019*.

Mingzhe Wang, Yihe Tang, Jian Wang, and Jia Deng. 2017a. Premise selection for theorem proving by deep graph embedding. *arXiv preprint arXiv:1709.09994*.

Qingxiang Wang, Chad Brown, Cezary Kaliszyk, and Josef Urban. 2020. Exploration of neural machine translation in autoformalization of mathematics in mizar. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 85–98. https://doi.org/10.1145/3372885.3373827

Yan Wang, Xiaojiang Liu, and Shuming Shi. 2017b. Deep neural solver for math word problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 845–854. https://doi.org/10.18653/v1/D17-1088

Zichao Wang, Andrew S. Lan, and Richard G. Baraniuk. 2021. Mathematical formula representation via tree embeddings. In *iTextbooks@ AIED*, pages 121–133. https://doi.org/10.1109/BigData52589.2021.9671942

Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hannaneh Hajishirzi, Yejin Choi, and Kyunghyun Cho. 2021a. Naturalproofs: Mathematical theorem proving in natural language. *arXiv preprint arXiv:2104.01112*.

Sean Welleck, Jiacheng Liu, Jesse Michael Han, and Yejin Choi. 2021b. Towards grounded natural language proof generation. In *MathAI4Ed Workshop at NeurIPS*.

Magdalena Wolska and Mihai Grigore. 2010. Symbol declarations in mathematical writing.

Magdalena Wolska and Ivana Kruijff-Korbayová. 2004. Analysis of mixed natural and symbolic input in mathematical dialogs. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 25–32. https://doi.org/10.3115/1218955.1218959

Yuhuai Wu, Albert Q. Jiang, Wenda Li, Markus N. Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. 2022. Autoformalization with large language models.

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

Zhipeng Xie and Shichao Sun. 2019. A goal-driven tree-structured neural model for math word problems. In *IJCAI*, pages 5299–5305. https://doi.org/10.24963/ijcai.2019/736

Ke Yuan, Dafang He, Zhuoren Jiang, Liangcai Gao, Zhi Tang, and C. Lee Giles. 2020. Automatic generation of headlines for online math questions. In *AAAI*, pages 9490–9497. https://doi.org/10.1609/aaai.v34i05.6493

Richard Zanibbi, Akiko Aizawa, Michael Kohlhase, Iadh Ounis, Goran Topic, and Kenny Davila. 2016a. Ntcir-12 mathir task overview. In *NTCIR*.

Richard Zanibbi and Dorothea Blostein. 2011. Recognition and retrieval of mathematical expressions. *International Journal on Document Analysis and Recognition (IJDAR)*, 15(4):331–357. https://doi.org/10.1007/s10032-011-0174-4

Richard Zanibbi, Kenny Davila, Andrew Kane, and Frank Tompa. 2015. The tangent search engine: Improved similarity metrics and scalability for math formula search. *arXiv preprint arXiv:1507.06235*. https://doi.org/10.1145/2911451.2911512

Richard Zanibbi, Kenny Davila, Andrew Kane, and Frank Wm Tompa. 2016b. Multi-stage math formula search: Using appearance-based similarity metrics at scale. In *Proceedings of the 39th International ACM SIGIR conference on*

Research and Development in Information Retrieval, pages 145–154. https://doi.org/10.1145/2911451.2911512

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. https://doi.org/10.1109/TPAMI.2019.2914054, PubMed: 31056490

Jipeng Zhang, Lei Wang, Roy Ka-Wei Lee, Yi Bin, Yan Wang, Jie Shao, and Ee-Peng Lim. 2020. Graph-to-tree learning for solving math word problems. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.362

Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. *arXiv preprint arXiv:1802.09691*.

Wenqi Zhang, Yongliang Shen, Yanna Ma, Xiaoxia Cheng, Zeqi Tan, Qingpeng Nong, and Weiming Lu. 2022a. Multi-view reasoning: Consistent contrastive learning for math word problem. *arXiv preprint arXiv:2210.11694*.

Yi Zhang, Guangyou Zhou, Zhiwen Xie, and Jimmy Xiangji Huang. 2022b. Hgen: Learning hierarchical heterogeneous graph encoding for math word problem solving. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. https://doi.org/10.1109/TASLP.2022.3145314

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. 2021. Minif2f: A cross-system benchmark for formal olympiad-level mathematics. *arXiv preprint arXiv:2109.00110*.

Wei Zhong, Shaurya Rohatgi, Jian Wu, C. Lee Giles, and Richard Zanibbi. 2020. Accelerating substructure similarity search for formula retrieval. In *European Conference on Information Retrieval*, pages 714–727. Springer. https://doi.org/10.1007/978-3-030-45439-5_47

Wei Zhong, Jheng-Hong Yang, and Jimmy Lin. 2022. Evaluating token-level and passage-level dense retrieval models for math information retrieval. *arXiv preprint arXiv:2203.11163*.

Wei Zhong and Richard Zanibbi. 2019. Structural similarity search for formulas using leaf-root paths in operator subtrees. In *European Conference on Information Retrieval*, pages 116–129. Springer. https://doi.org/10.1007/978-3-030-31624-2

Claus Zinn. 1999. Understanding mathematical discourse. In *Proceedings of the Workshop on the Semantics and Pragmatics of Dialogue, Amsterdam University*. Citeseer.

Claus Zinn. 2003. A computational framework for understanding mathematical discoursexy. *Logic Journal of IGPL*, 11(4):457–484. https://doi.org/10.1093/jigpal/11.4.457

A Approach-specific Limitations

Identifier-Definition Extraction Limitations. Methods considering the link between identifiers and their definitions have split off into at least three recent tasks: identifier-definition extraction (Schubotz et al., 2017; Alexeeva et al., 2020), variable typing (Stathopoulos et al., 2018), and notation auto-suggestion (Jo et al., 2021). A lack of consensus on the framing of the task and data prevents a direct comparison between methods. Schubotz et al. (2017) advise against using their gold standard data for training due to certain extractions being too difficult for automated systems, among other reasons. They also propose future research should focus on recall due to current methods extracting exact definitions for only 1/3 of identifiers, and suggest use of multilingual semantic role labeling (Akbik et al., 2016) and logical deduction (Schubotz et al., 2016b). Logical deduction is partially tackled by Alexeeva et al. (2020), which is based on an open-domain causal IE system (Sharp et al., 2019) with Odin grammar (Valenzuela-Escárcega et al., 2016), where temporal logic is used to obtain intervals referred to by pre-identified time expressions (Sharp et al., 2019). We assume the issues with superscript identifiers (such as Einstein notation, *etc.*) from Schubotz et al. (2016b) carry over into Schubotz et al. (2017). The rule-based approach proposed by Alexeeva et al. (2020) attempts to account for such notation (known as *wildcards* in formula retrieval). They propose that future methods should combine grammar with a learning framework,

extend rule sets to account for coordinate constructions, and create well-annotated training data using tools such as PDFAlign and others (Asakura et al., 2021).

Formula Retrieval Limitations. Zhong and Zanibbi (2019) propose supporting query expansion of math synonyms to improve recall, and note that Approach0 does not support wildcard queries. Zhong et al. (2020) later provide basic support for wildcards. Tangent-CFT also does not evaluate on wildcard queries, and the authors suggest extending the test selection to include more diverse formulae, particularly those that are not present as exact matches. They propose integrating nearby text into learned embeddings. MathBERT (Peng et al., 2021) performs such integration, but does not learn *n*-gram embeddings. MathBERT evaluates on non-wildcard queries only.

Informal Premise Selection Limitations. Limitations involve a lack of structural consideration of formulae and limited variable typing abilities. Ferreira and Freitas (2020b) note that the graph-based approach to premise selection as link prediction struggles to encode mathematical statements which are mostly formulae, and suggest inclusion of structural embeddings (*e.g.*, MathBERT [Peng et al., 2021]) and training BERT on a mathematical corpus. They also describe value in formulating sophisticated heuristics for navigating the premises graph. Later, following a Siamese network architecture (Ferreira and Freitas, 2021) reliant on dual-layer word/expression self-attention and a BiLSTM (STAR), the authors demonstrate that STAR does not appropriately encode the semantics of variables. They suggest that variable typing and representation are a fundamental component of encoding mathematical statements. Han et al. (2021) plan to explore the effect of varying pre-training components, testing zero-shot performance without contrastive fine-tuning, and unsupervised retrieval. Coavoux and Cohen (2021) propose a statement-proof matching task akin to informal premise selection, with a solution reliant on a self-attentive encoder and bilinear similarity function. The authors note model confusion due to the proofs introducing new concepts and variables rather than referring to existing concepts.

Math Word Problem Limitations. In Graph2-Tree-Z, Zhang et al. (2020) suggest considering more complex relations between quantities and language, and introducing heuristics to improve solution expression generation from the tree-based decoder. In EPT, Kim et al. (2020) find error probability related to fragmentation issues increases exponentially with number of unknowns, and propose generalizing EPT to other MWP datasets. HGEN (Zhang et al., 2022b) note three areas of future improvement: Combining models into a unified framework through ensembling multiple encoders (similar to Ferreira and Freitas, 2021); integrating external knowledge sources (*e.g.*, HowNet (Dong and Dong, 2003), Cilin (Hong-Minh and Smith, 2008)); and real-world dataset development for unsupervised or weakly supervised approaches (Qin et al., 2020).

Informal Theorem Proving Limitations. Wang et al. (2020) suggest the development of high-quality datasets for evaluating translation models, including structural formula representations, and jointly embedding multiple proof assistant libraries to increase formal dataset size. Szegedy (2020) argues that reasoning systems based on self-driven exploration without informal communication abilities would suffer usage and evaluation difficulties. Wu et al. (2022) note limitations with text window size and difficulty storing large formal theories with current language models. After proposing the NaturalProofs dataset, Welleck et al. (2021a) characterize error types for the full-proof generation and next-step suggestion tasks, noting issues with: (1) hallucinated references, meaning the reference does not occur in NaturalProofs; (2) non-ground-truth reference, meaning the reference does not occur in the ground-truth proof; (3) undefined terms; and (4) improper or irrelevant statement, meaning a statement that is mathematically invalid (*e.g.*, $2/3 \in \mathbb{Z}$) or irrelevant to the proof; and (5) statements that do not follow logically from the preceding statements. Dealing with research-level physics, Meadows and Freitas (2021) note the significant cost of semi-automated formalization, requiring detailed expert-level manual intervention. They also call for a set of well-defined computer algebra operations such that robust mathematical exploration can be guided in a goal-based setting.

B Diagrammatic Categorization of Approaches

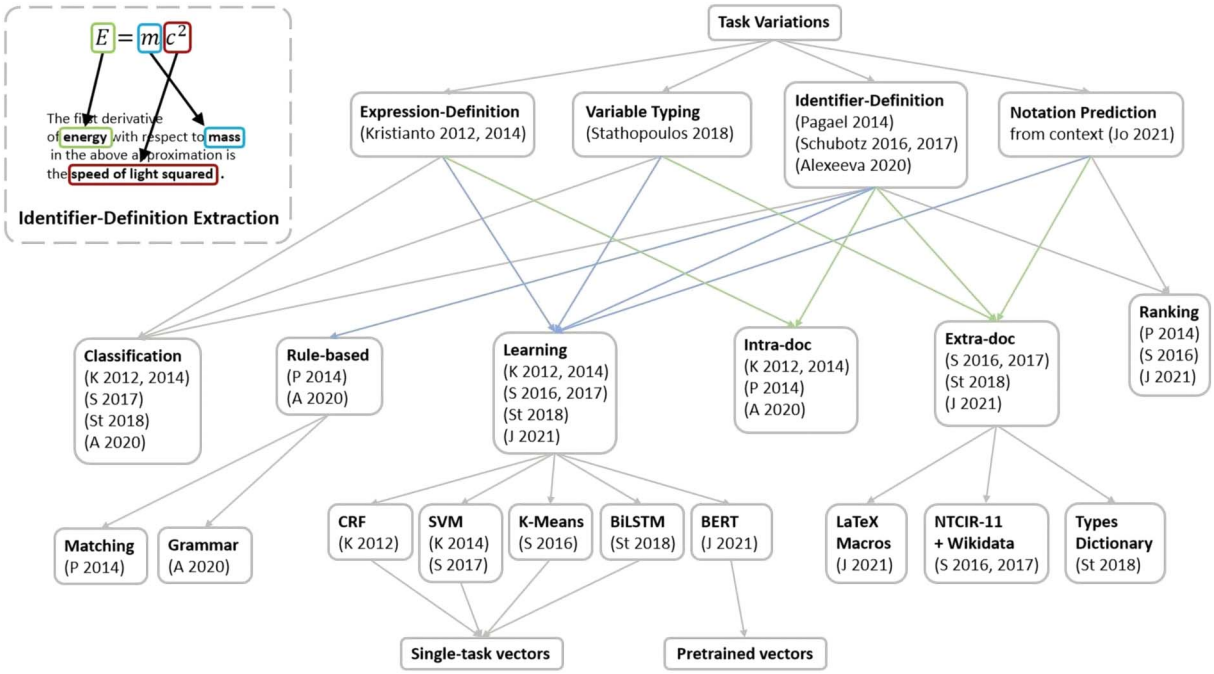

Figure 3: Categorisation of approaches related to identifier-definition extraction. The shorthand notation used such as (K 2012) and (J 2021) refer to the references in the first four boxes, *i.e.*, (Kristianto 2012) and (Jo 2021). The first four boxes are task variations, then arrows point to other categories that may group approaches. For example, (Stathopoulos 2018) is *Variable Typing*, considers a *Classification* task, involves a large machine *Learning* element, uses a *BiLSTM*, learns *Vector* representations of input text without pretraining, and relies on information outside of the instance text (*Extra-doc*), which is a *Types Dictionary*.

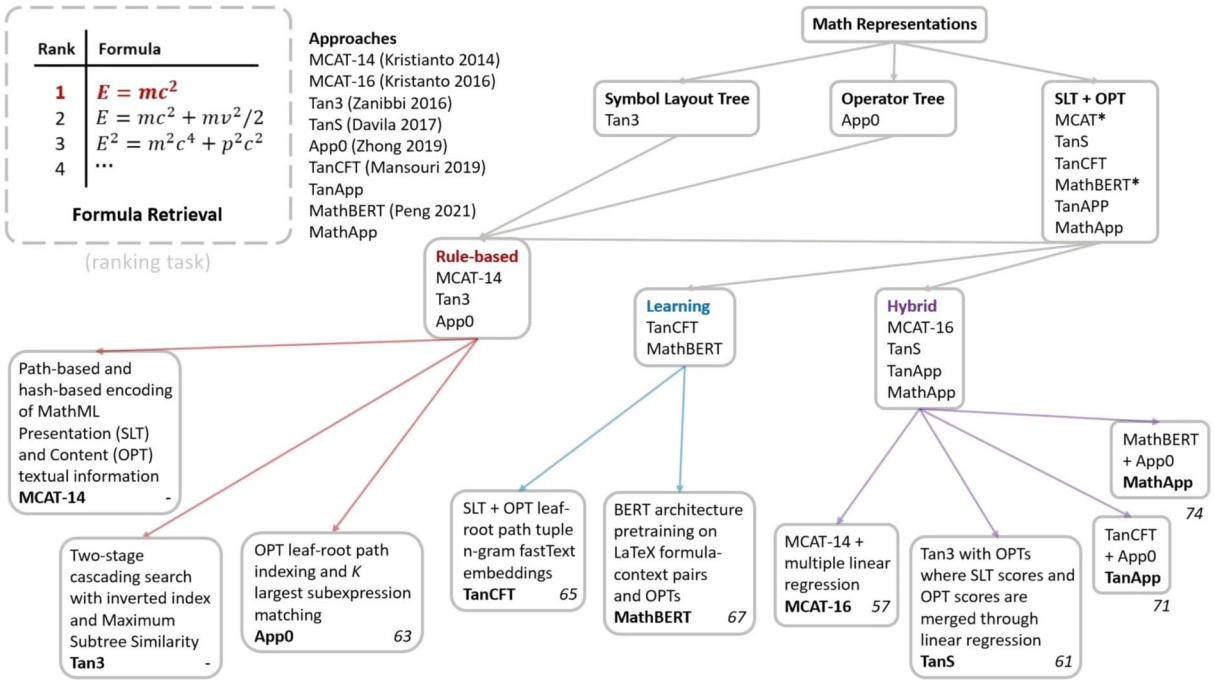

Figure 4: Categorisation of approaches in formula retrieval. The number at the bottom right of boxes refers to their respective Bpref score (Peng et al., 2021).

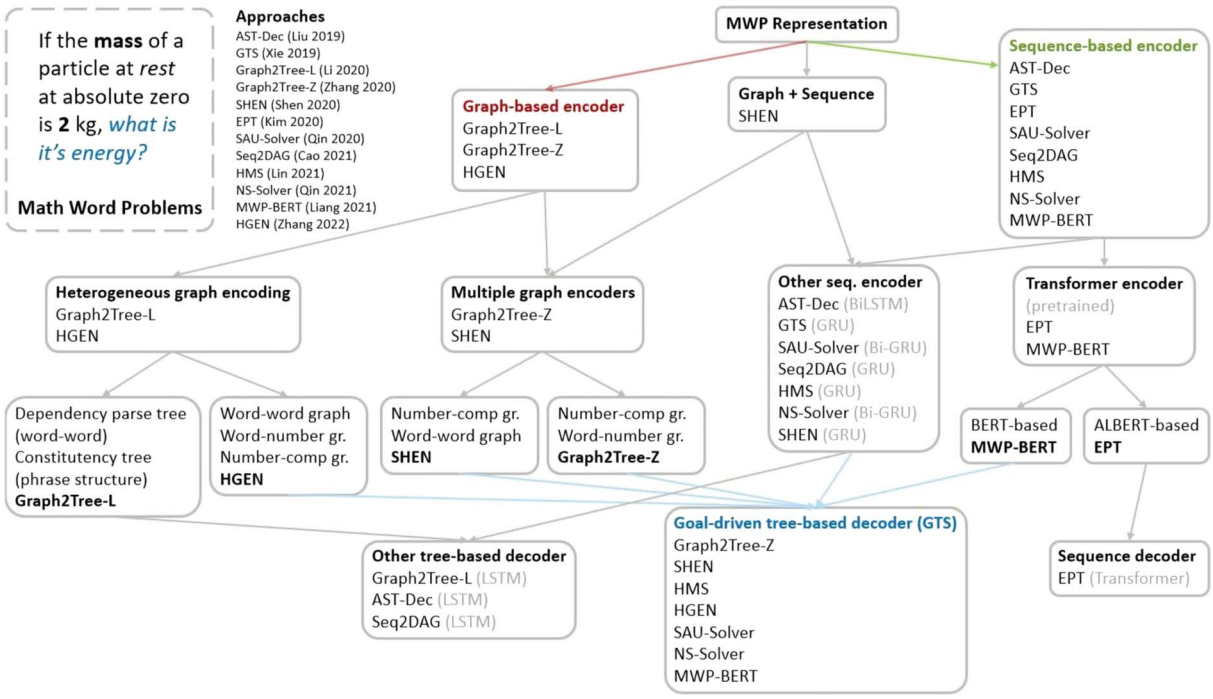

Figure 5: Categorisation of approaches in math word problem solving.