Why are NLP Models Fumbling at Elementary Math? A Survey of Automatic Word Problem Solvers

Anonymous ACL submission

Abstract

From the latter half of the last decade, there has been growing interest in developing algorithms for automatically solving mathematical word problems (MWP). It is an exciting language problem which demands not only surface level text pattern recognition but requires coupling with mathematical reasoning as well. In spite of the dedicated effort, we are still miles away from building robust representations of elementary math word problems. In this paper, we critically examine the various models that have been developed for solving word problems, their pros and cons and the challenges ahead. In the last two years, a lot of deep learning models have come out with competing results on benchmark datasets. We take a step back and analyse why, in spite of this, the predominantly used experiment and dataset designs are a stumbling block and provide a road-map for the future.

1 Introduction

Natural language processing has been one of the most popular and intriguing AI-complete sub-fields of artificial intelligence. One of the earliest systems arguably was the PhD Thesis on automatically solving arithmetic word problems (Bobrow, 1964). The challenge lay on two fronts (a) analysing unconstrained natural language and (b) mapping infinite text patterns onto a small mathematical vocabulary and reasoning framework.

Right up until 2010, there has been prolific exploration of MWP solvers, for various domains (such as algebra, percentages, ratio etc). These solvers relied heavily on hand-crafted rules for bridging the gap between language and the corresponding mathematical notation. As can be surmised, these approaches did not generalise well. Moreover, due to the lack of well accepted datasets, it is hard to measure the relative performance across proposed systems (Mukherjee and Garain, 2008).

Input	Kevin has 3 books. Kylie has 7 books. How many books do they have together?
Answer	10

Table 1: Typical Example

The pioneering work by (Kushman et al., 2014) employed statistical methods to solve word problems, which set the stage for the development of automatic MWP solvers using traditional machine learning methods. The work also introduced the first dataset, popularly referred to as Alg514, that used multiple linear equations for solving the problem. The machine learning model mapped the coefficients in the equation to the numbers in the problem. Hence, the dataset comprised of the natural language question, equation set and the final answer.

Mirroring recent trends in NLP, there has been an explosion of deep learning models for MWP. Some of the early ones (Wang et al., 2017; Ling et al., 2017) modeled the task of converting the text to equation as a Seq2Seq problem. In this context, increasingly complex models have been proposed to capture semantics beyond the surface text. Some have captured structural information (pertaining to input text, domain knowledge, output equation structure) in the form of graphs and use advances in graph neural networks ((Li et al., 2020), (Zhang et al., 2020c), etc.). Others have utilised the benefits of transformers in their modelling ((Liang et al., 2021), (Piękos et al., 2021), etc.). We will explore these models in detail.

Since this is a problem that has consistently attracted attention, ostensibly right from the birth of the field of NLP, a survey of the problem solving techniques offers a good horizon for researchers. In the last three years, the authors were able to collect 30+ papers on deep learning for word problem solving, presented at premier NLP venues. Each paper has its own unique intuition and achieves similar performances. These kind of parallel publications has made it hard to ascertain what are the State-of-the-Art (SOTA) results. The way the research has progressed at break-neck speed has caused a clustering of models around similar performance values. Hence, a broad overview of the techniques employed gives a good grounding for further research. Similarly, understanding the source, settings and relevance of datasets is important. For example, there are many datasets that are often referred to by multiple names at different points in time. Also, the problem setting varies across systems (whether multiple equations can be solved, whether it is restricted to algebra or more domains etc.) In this survey, we systematically analyse the models, list the benchmark datasets and examine them thoroughly under a critical lens.

1.1 Related Surveys

There are two seminal surveys that are cited in this field. One, (Mukherjee and Garain, 2008), has a detailed overview of the symbolic solvers for this problem. The second, more recent one (Zhang et al., 2020a), covers models proposed up until 2020. In the last two years, there has been a sharp spike in algorithms developed, that focus on various aspects of deep learning, to model this problem. Our survey is predominantly based on these deep learning models, and takes a hard-look at why they are often brittle, and how that is a symptom of deficient dataset design, as well as model design, and finish with some directions for mitigating the same in future.

2 Symbolic Solvers

We begin our discussion with traditional solvers that employ a rule-based method to convert text input to a set of *symbols*. Starting with STUDENT (Bobrow, 1964) program, and other attempts ((Fletcher, 1985), (Dellarosa, 1986), (Bakman, 2007), etc. mapped the natural language input to an underlying pre-defined *schema*, i.e., a mechanism that identifies common expectations of language, word problems and the corresponding mathematical notation. Commonly, this involved setting up a slot-filling mechanisms that mapped the main entities of the word problem to a set of equations. An example of a schema for algebraic MWP is shown in Table 2.

Problem	John has 5 apples. He gave 2 to Mary. How many does he have now?
Template	[Owner$_1$] has [X] [obj]. [Owner$_1$] [transfer] [Y] [obj] to [Owner$_2$]. [Owner$_1$] has [Z] [obj]. Z = X - Y
Slot-Filling	[John] has [5] [apple]. [John] [give] [2] [apple] to [Mary]. [Mary] has [Z] [apple]. Z = 5 - 2
Answer	Z = 3

Table 2: Workflow of Symbolic Solvers

The advantage is that these systems are robust in handling irrelevant information. In addition, multiple works were proposed to customize these symbolic systems for various domains (Mukherjee and Garain, 2008). As one can observe, the rules would need to be exhaustive to capture the myriad nuances of language. Moreover, they did not generalise well on the language front. Since each system was designed for a particular mathematical complexity of solving, without the use of datasets, it was difficult to evaluate performance across systems.

3 Statistical Solvers

As with many tasks in natural language processing, statistical machine learning techniques to solve word problems started dominating the field from 2014. The central theme of these algorithms is to score a number of potential solutions (may be equations or expression trees as we will see presently) as an optimisation problem, and subsequently arrive at the correct mathematical model for the given text. This poses the problem as a **structure prediction problem**.

$$P(y|x;\theta) = \frac{e^{\theta.\phi(x,y)}}{\sum_{y' \in Y} e^{\theta.\phi(x,y')}} \qquad (1)$$

As with optimization problems, Equation 1 refers to the problem of learning parameters θ, which relate to the feature function ϕ. Consider labeled dataset D consisting of n pairs (x, y, a) where x is the natural language question, y is the mathematical expression and a is the numerical answer. The task is to score all possible expressions Y, and maximise the choice of the labelled y through an optimisation setting. This is done by modifying the parameters θ of the feature function

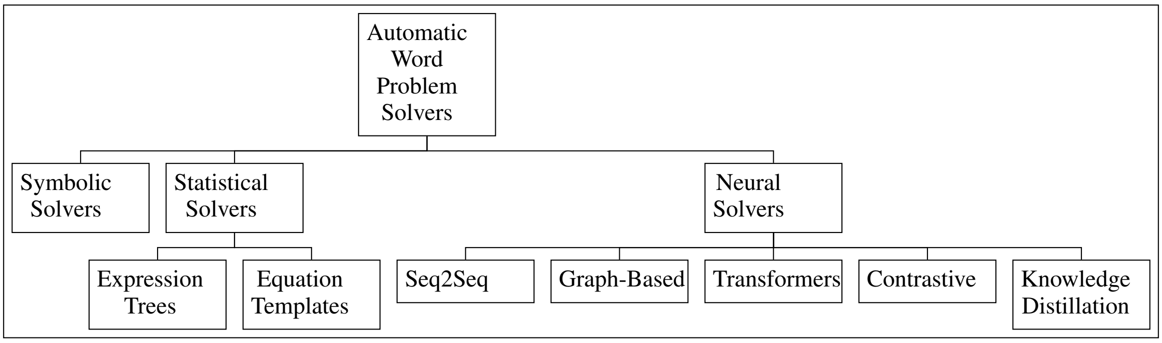

Figure 1: Types of Word Problem Solvers

$\phi(x, y)$. Different models propose different formulations of ϕ. In practise, beam search is used as a control mechanism. We grouped the prolific algorithms that were developed, based on the type of mathematical structure y - either as *equation templates* or *expression trees*. Equation templates were mined from training data, much like the slot filling idea of symbolic systems. However, they became a bottleneck to generalizability, if the word problem at inference time, was from an unseen equation template. To address this issue, expression trees, with unambiguous post-fix traversals, were used to model equations. Though they restricted the complexity of the systems to single equation models, they offered wider scope for generalizability.

3.1 Equation Templates

To begin with, (Kushman et al., 2014), used structure prediction to score both equation templates and alignment of the numerals in the input text to coefficients in the template. Using a state based representation, (Hosseini et al., 2014) modelled simple elementary level word problems with emphasis on verb categorisation. (Zhou et al., 2015) enhanced the work done by (Kushman et al., 2014) by using quadratic programming to increase efficiency. (Upadhyay and Chang, 2017) introduced a sophisticated method of representing derivations in this space.

3.2 Expression Trees

Expression tree based methods converge faster, understandably due to the diminished complexity of the model. Some solvers (such as (Roy and Roth, 2016)) had a joint optimisation objective to identify relevant numbers and populating the expression tree. On the other hand, (Koncel-Kedziorski et al., 2015; Mitra and Baral, 2016) used domain knowledge to constrain the search space.

4 Neural Solvers

The solvers described until now, involved an overhead of converting the input text into the feature space through a myriad of ways. With the advent of distributed representations for text (Le and Mikolov, 2014; Peters et al., 2018; Pennington et al., 2014; Devlin et al., 2018), including domain specific ones like (Sundaram et al., 2020), deep learning algorithms entered the fray. Starting with (Ling et al., 2017), which designed a Seq2Seq model that incorporated learning a program (as an intermediate step) as well, there has been a flurry of activity. Almost all the deep learning solvers model the problem as a language translation task, i.e., translating from the input natural language text to a sequence of characters representing either the equation or a sequence of predicates. This design choice has severely restricted the choice of mathematical problems that can be attempted by this architecture. To illustrate, equation systems that involve solving multiple equations are not modeled well. A notable exception to this is the popular baseline MathDQN (Wang et al., 2018), which employs deep reinforcement learning.

4.1 Seq2Seq Solvers

The ubiquitous Seq2Seq ((Sutskever et al., 2014)) architecture is widely popular for automatic word problem solving. From early direct use of LSTMs (Hochreiter and Schmidhuber, 1997) / GRUs (Cho et al., 2014) in Seq2Seq models ((Huang et al., 2017), (Wang et al., 2017)) to complex models that include domain knowledge ((Ling et al., 2017), , , (Chatterjee et al., 2021), (Qin et al., 2020), (Chiang and Chen, 2019), (Qin et al., 2021) etc.), diverse formulations of this basic architecture have been employed.

4.2 Graph-based Solvers

With the advent of graph modeling (Xia et al., 2019) and the scope of multi-modal processing, the graph became a vehicle for adding knowledge to solvers. One way of doing that was to model the input problem as a graph ((Feng et al., 2021), (Li et al., 2020), (Yu et al., 2021), (Hong et al., 2021)). This incorporates domain knowledge of (a) language interactions pertinent to mathematical reasoning or (b) quantity graphs stating how various numerals in the text are connected. Another way is to model the decoder side to accept graphical input of equations ((Xie and Sun, 2019), (Lin et al., 2021), (Zaporojets et al., 2021), (Cao et al., 2021), (Liu et al., 2019), (Wu et al., 2021b)). The natural extension is to use graph neural networks for both encoder and decoder ((Zhang et al., 2020c), (Wu et al., 2020), (Wu et al., 2021a), (Shen and Jin, 2020)).

4.3 Transformers

Transformers (Vaswani et al., 2017) have revolutionised the field of NLP. Word problem solving has been no exception. Through the use of BERT (Devlin et al., 2018) embeddings or through transformer based encoder-decoder models, a few systems use concepts from transformers ((Liu et al., 2019), (Kim et al., 2020)) . In some cases, the translation is from text to explanation ((Piękos et al., 2021), (Griffith and Kalita, 2020)), or from text to equation ((Shen et al., 2021), (Liang et al., 2021)).

4.4 Contrastive Solvers

With the introduction of Siamese networks, (Koch et al., 2015), the idea of building representations that *contrast* between vectorial representations, that reflect that contrast between various classes in data. In the context of word problem solving, a few transformer based encoder-decoder models ((Li et al., 2021), (Hong et al., 2021)) have been proposed, that utilize the concept of contrastive learning (Le-Khac et al., 2020).

4.5 Teacher-Student Solvers

The paradigm of knowledge distillation, in the wake of large, generic end-to-end models, has become immensely popular (add citation). Since word problem datasets are of comparatively smaller size, it is but logical that large generic networks can be fine-tuned for downstream processing of word problem solving, as favourably demonstrated by (Zhang et al., 2020b) and (Hong et al., 2021).

5 Domain-Niche Solvers

A small set of works, amongst both statistical solvers and deep models, focus on the pertinent characteristics of a particular domain in mathematics, such as probability word problems (Dries et al., 2017; Suster et al., 2021; Tsai et al., 2021), number theory word problems (Shi et al., 2015), geometry word problems (Seo et al., 2015; Chen et al., 2021), age word problems (Sundaram and Abraham, 2019) and so on.

6 Datasets

Datasets used for math word problem solving are listed in Table 3 with their characteristics. The top section of the table describes datasets with relatively less number of problems, sufficient for algorithm that employed statistical learning techniques. The bottom half consists of more recent datasets that are more suitable for deep learning algorithms.

6.1 Small Datasets

The pioneering work in solving word problems (Kushman et al., 2014), introduced a comprehensive dataset (Alg514) of 514 word problems, across various domains in algebra (such as percentages, mixtures, speeds etc). This dataset was annotated with multiple equations per problem. AddSub was introduced in (Hosseini et al., 2014), with simple addition/subtraction problems, with limited language complexity. SingleOp (Roy et al., 2015) and MultiArith (Roy and Roth, 2016) were proposed such that there is a control over the operators (single operator in the former and two operators in the latter). SingleEq (Koncel-Kedziorski et al., 2015) is an interesting dataset, which incorporates long sentence structures for elementary level school problems. AllArith (Roy and Roth, 2017) is a subset of the union of AddSub, SingleEq and SingleOp. "Perturb" is a set of slightly perturbed word problems of AllArith and finally Aggregate is the union of AllArith and Perturb. MAWPS (A **Ma**th **W**ord **P**roblem **S**olving Repository) (Koncel-Kedziorski et al., 2016) is a curated dataset (with deliberate template overlap control) that is comprised of all the proposed datasets till that date. A single equation subset of MAWPS has been studied (Miao et al., 2021), for diagnostic analysis of solvers. Similarly, the critique offered by (Patel et al., 2021) was demonstrated using their newly proposed dataset SVAMP. All the mentioned datasets are annotated

Dataset	Type	Domain	Size	Source
Alg514 (SimulEq-S)	Multi-equation	(+,-,*,/)	514	(Kushman et al., 2014)
AddSub (AI2)	Single-equation	(+,-)	340	(Hosseini et al., 2014)
SingleOp (Illinois, IL)	Single-equation	(+,-,*,/)	562	(Roy et al., 2015)
SingleEq	Single-equation	(+,-,*,/)	508	(Koncel-Kedziorski et al., 2015)
MAWPS	Multi-equation	(+,-,*,/)	3320	(Koncel-Kedziorski et al., 2016)
MultiArith (Common Core, CC)	Single-equation	(+,-,*,/)	600	(Roy and Roth, 2016)
AllArith	Single-equation	(+,-,*,/)	831	(Roy and Roth, 2017)
Perturb	Single-equation	(+,-,*,/)	661	(Roy and Roth, 2017)
Aggregate	Single-equation	(+,-,*,/)	1492	(Roy and Roth, 2017)
DRAW-1k	Multi-equation	(+,-,*,/)	1k	(Upadhyay and Chang, 2017)
AsDIV-A	Single-equation	(+,-,*,/)	2373	(Miao et al., 2021)
SVAMP	Single-equation	(+,-,*,/)	1000	(Patel et al., 2021)
Dolphin18k	Multi-equation	(+,-,*,/)	18k	(Huang et al., 2016)
AQuA-RAT	Multiple-choice	-	100k	(Ling et al., 2017)
Math23k*	Single-equation	(+,-,*,/)	23k	(Huang et al., 2017)
MathQA	Single-equation	(+,-,*,/)	35k	(Amini et al., 2019)
HMWP*	Multi-equation	(+,-,*,/)	5k	(Qin et al., 2020)
Ape210k*	Single-equation	(+,-,*,/)	210k	(Liang et al., 2021)
GSM8k	Single-equation	(+,-,*,/)	8.5k	(Cobbe et al., 2021)
EW10k	Single-equation	(+,-,*,/)	10k	(Chatterjee et al., 2021)
CM17k*	Multi-equation	(+,-,*,/)	17k	(Qin et al., 2021)

Table 3: Datasets
(*Chinese Datasets)

with both the *equation* and the *answer*. While running experiments and creating cross-validation sets, one must keep in mind various subsets and supersets.

6.2 Large Datasets

Dolphin18k (Huang et al., 2016) is an early proprietary dataset that was evaluated primarily with the statistical solvers. AQuA-RAT (Ling et al., 2017) introduced the first large crowd-sourced dataset for word problems with *rationales* or *explanations*. The setting is quite different from the aforementioned datasets, not only with respect to size, but also in the wide variety of domain areas (spanning physics, algebra, geometry, probability etc). Another point of difference is that, the annotation involves the entire textual explanation, rather than equations alone. MathQA (Amini et al., 2019) critically analysed AQuA-RAT and selected the core subset and annotated it with a predicate list. Once again, care must be taken that MathQA is a subset of AQuA-RAT. GSM8k (Cobbe et al., 2021) is a recent single-equation dataset, that is the large scale version of AsDIV-A (Miao et al., 2021). Math23K is a popular Chinese dataset for single equation math word problem solving. A recent successor is Ape210k (Liang et al., 2021).

7 Performance of Deep Models

In this section, we describe the performance of neural solvers.

Evaluation Measures: The most popular metric is *answer accuracy*, which evaluates the predicted equation and checks whether it is the same as the labelled one. The other metric is *equation accuracy*, which predominantly does string matching and matches the equation to the annotated equations.

We have listed the performance of the deep models in Table 4, on two major datasets - Math23K and MAWPS. Some of these deep models report scores on other datasets as well. For conciseness, we have chosen the most popular datasets for deep models. We see that, in general, the models achieve around 70-80 percentage points on *answer accuracy*. (Shen et al., 2021) outperforms all other models on Math23k whereas RPKHS (Yu et al., 2021) is the best model for MAWPS till date. Apart from these algebraic datasets, multi-domain datasets MathQA and AquA are also of special interest. This is described in Table 5. The interesting takeaway is that, the addition of BERT modelling to AQuA (Piękos et al., 2021), still performed slightly worse than the Seq2Prog (Amini et al., 2019) model, which is a derivative of the Seq2Seq paradigm. This suggests that while the results are commendable, a closer look reveals that there is much scope for improving word problem modelling.

Model Name	Math23k	MAWPS	Source
GTS	74.3	-	(Xie and Sun, 2019)
SAU-SOLVER	74.8	-	(Chiang and Chen, 2019)
Group-att	69.5	76.1	(Li et al., 2019)
Graph2Tree	77.4	-	(Li et al., 2020)
KA-S2T	76.3	-	(Wu et al., 2020)
NS-Solver	75.67	-	(Qin et al., 2020)
Graph-To-Tree	78.8	-	(Li et al., 2020)
TSN-MD	77.4	84.4	(Zhang et al., 2020b)
Graph-To-Tree+Teacher	79.1	84.2	(Liang and Zhang, 2021)
NumS2T	78.1	-	(Wu et al., 2020)
Multi-E/D	78.4	-	(Shen and Jin, 2020)
EPT	-	84.5	(Kim et al., 2020)
Seq2DAG	77.1	-	(Cao et al., 2021)
WARM	80.1	-	(Chatterjee et al., 2021)
EEH-D2T	78.5	84.8	(Wu et al., 2021a)
Generate and Rank	**85.4**	84.0	(Shen et al., 2021)
HMS	76.1	80.3	(Lin et al., 2021)
RPKHS	83.9	**89.8**	(Yu et al., 2021)
CL	83.2	-	(Li et al., 2021)
GTS+RODA	77.9	-	(Liu et al., 2022)

Table 4: Answer Accuracy of Deep Models

8 Analysis of Deep Models

In this section of the paper, we analyze the pros and cons of applying deep learning techniques to solve word problems automatically. At the outset, two layers of understanding are imperative (i) linguistic structures that describe a situation or a sequence of events and (ii) mathematical structures that govern these language descriptions. Though deep learning models have rapidly scaled and demonstrated commendable results for capturing these characteristics, when one examines the problem more closely, a plethora of insights are available for further exploration. The predominant modus-operandus is to create a deep model that converts the input natural language to the underlying equation. In some cases, the input is converted into a set of predicates (Amini et al., 2019) or explanations (Ling et al., 2017).

Model	AQuA-RAT	MathQA	Source
AQuA	36.4	-	(Ling et al., 2017)
Seq2Prog	**37.9**	57.2	(Amini et al., 2019)
BERT-NPROP	37.0	-	(Piękos et al., 2021)
Graph-To-Tree	-	**69.65**	(Li et al., 2020)

Table 5: Performance on Large Multi-Domain Datasets

8.1 What Shortcuts are being Learned?

Shortcut Learning (Geirhos et al., 2020) is a recently well-studied phenomenon of deep neural networks. It describes how deep learning models learn patterns in a shallow way and fall prey to questionable generalizations across datasets (an example is an image being classified as sheep if there was grass alone; due to peculiarities in the dataset). This is a function of the low-level input we provide to such models (pixels, word embeddings etc.). In the context of word problems, (Patel et al., 2021) exposed how removing the question and simply passing the situational context, leads to the correct equation being predicted. This suggests two things, issues with model design as well as issues with dataset design. The datasets have high equation template overlap, as well as text overlap. Word problem solving is a hard because two otherwise identical word problems, with a small word change (say changing the word *give* to *take*), would completely change the equation. Hence high lexical similarity does not translate to corresponding similarity in the mathematical realm (Patel et al., 2021; Sundaram et al., 2020).

8.2 Is Language or Math being Learned?

The question that looms large is whether adequate mapping of language to math has been modelled, whether linguistic modelling has been unfavourably highlighted or that the mathematical aspects have been captured succinctly. We claim that both language and math have not yet been modelled ade-

Problem	Solved?
John has 5 apples. Mary has 2 apples more than John. How many apples does Mary have?	Yes
John has 5 apples. Mary has 2 apples more than John. Who has less apples?	No
What should be added to two to make it five?	No

Table 6: Behaviour of Baseline BERT Model

quately. Apart from the perturbations experiment done by SVAMP (Patel et al., 2021), which exposes that the mapping between linguistic and mathematical structures is not captured, we suggest two more experiments that expose flaws in linguistic and mathematical modelling alone. The first one involves imposing a question answering task on top of the word problem as a probing test. For example, a baseline BERT model that converts from input language to equation (Table 6), trained on MAWPS, can solve a simple word problem such as "*John has 5 apples. Mary has 2 apples more than John. How many apples does Mary have?*", but cannot answer the following allied question "*John has 5 apples. Mary has 2 apples more than John. Who has less apples?*". One reason is of course, dataset design. The governing equation for this problem is "X = 5-2". However, the text version of this, "*What should be added to two to make it five?*", cannot be solved by the baseline model. Similarly, many solvers wrongly output equations such as "X = 2 - 5" (Patel et al., 2021), which suggests mathematical modelling of subtraction of whole numbers is not up to the mark. Hence, we observe, that deep translation models neither model language, nor the math sufficiently.

8.3 Is Accuracy Enough?

As suggested by the discussion above, a natural line of investigation is to examine the evaluation measures, and perhaps the error measures for the deep models, in order to bring about a closer coupling between syntax and semantics. High accuracy of the models to predicting the answer or the equation suggests a shallow mapping between the text and the mathematical symbols. One direction of exploration is data augmentation with a single word problem annotated with multiple equivalent equations. Metrics that measure the soundness of the equations generated, the robustness of the model to simple perturbations (perhaps achieved using a denoising autoencoder) and the ability of the model to discern important entities in a word problem (perhaps using an attention analysis based metric), are the need of the future. An endeavour has been done by (Kumar et al., 2021), where adversarial examples have been generated and utilised to evaluate SOTA models.

8.4 Are the Trained Models Reproducible?

Most of the SOTA systems come with their own, well-documented repositories. Though an aggregated toolkit (Lan et al., 2021) (open-source MIT License) is available, running saved models in inference mode, to probe the quality of the datasets, proved to be a hard task, with varying missing hyper-parameters or missing saved models. This, however, interestingly suggests that API's that can take a single word problem as input and computes the output, would be highly useful for application designers. This has been done in the earlier systems such as (Roy and Roth, 2018) and (Wolfram, 2015).

9 Analysis of Benchmark Datasets

In this section of the paper, we explore the various dimensions of the popular datasets (Table 3).

9.1 Low Resource Setting

Compared to usual text related tasks, the available datasets are quite small in size. They also suffer from a large lexical overlap (Amini et al., 2019). This taxes algorithms, that now have to generalise from an effectively small dataset.

9.2 Annotation Cost

The datasets currently have little to no annotation costs involved as they are usually scrapped from homework websites. There are some exceptions that involve crowd-sourcing (Ling et al., 2017) or intermediate representations apart from equations (Amini et al., 2019). Some efforts include removing the need for basic equation annotation, and relying only on the answer (Chatterjee et al., 2021).

9.3 Template Overlap

Many studies (Zhang et al., 2020a) have demonstrated that there is a high lexical and mathematical overlap between the word problems in popular datasets. Consequently, many strategies have been adopted to mitigate this. Early attempts include controlling linguistic and equation template overlap ((Koncel-Kedziorski et al., 2016), (Miao et al., 2021)). Later ideas revolve around controlled design and quality control of crowd-sourcing (Amini et al., 2019).

10 Road Ahead

In this section, we describe exciting frontiers of research for word problem solving algorithms.

10.1 Semantic Parsing

As rightly suggested by (Zhang et al., 2020a), the closest natural language task for word problem solving is that of *semantic parsing*, and not *translation* as most of the deep learning models have modelled. The mapping between extremely long chunks of text to short equation sentences has the advantage of generalising on the decoder side, but equally has the danger of overloading many involved semantics into a simplistic equation model. To illustrate, an equation may be derived after applying a sequence of steps that is lost in a simple translation process. A lot of efforts have already been employed in adding such a nuance. One way is to model the input intelligently ((Peng et al., 2021), (Liang et al., 2021)). The intermediate representations include simple predicates (Roy and Roth, 2018), while others involve a programmatic description ((Ling et al., 2017), (Amini et al., 2019)). Yet another way is to include semantic information in the form of graphs as shown in ((Huang et al., 2018), (Chiang and Chen, 2019), (Qin et al., 2020), (Li et al., 2020), etc.)).

10.2 Informed Dataset Design

As most datasets are scraped from websites, there is bound to be repetition. Some effort, if invested into designing datasets that expose (a) different versions of the same problem, (b) different equivalent equation types, (c) semantics of the language and the math. A step in this direction has been explored by (Patel et al., 2021), which provides a challenge dataset for evaluating word problems, and (Kumar et al., 2021) where adversarial examples are automatically generated.

10.2.1 Dataset Augmentation

A natural extension of dataset design, is dataset augmentation. Augmentation is a natural choice when we have datasets that are small and focused on a single domain. Then, linguistic and mathematical augmentation can be automated by domain experts. While template overlap is a concern in dataset design, it can be leveraged in contrastive designs as in ((Sundaram et al., 2020), (Li et al., 2021)). A principled approach of reversing operators and building equivalent expression trees for augmentation has been explored here (Liu et al., 2022).

10.2.2 Few Shot Learning

This is useful if we have a large number of non-annotated word problems or if we can come up with complex annotations (that capture semantics) for a small set of word problems. In this way *few shot learning* can generalise from few annotated examples.

10.3 Knowledge Aware Models

We propose that word problem solving is more involved than even semantic parsing. From an intuitive space, we learn language from examples and interactions but we need to be explicitly *trained* in math to solve word problems (Marshall, 1996). This suggests we need to include mathematical models into our deep learning models to build generalisability and robustness. As mentioned before, a common approach is to include domain knowledge as a graph ((Chiang and Chen, 2019), (Wu et al., 2020), (Qin et al., 2020), (Qin et al., 2021)).

11 Conclusion

In this paper, we surveyed the existing math word problem solvers, with a special focus on deep learning models. Deep models are predominantly modeled as encoder-decoder models, with input as text and decoder output as equations. We listed several interesting formulations of this paradigm - namely as Seq2Seq models, graph-based models, transformer-based models, contrastive models and teacher-student models. We then explored in detail the various datasets in use. Subsequently, we analysed the various approaches of modelling word problem solving, followed by the characteristics of the popular datasets. We concluded that the brittleness of the SOTA models was due to (a) tough modelling decisions and (b) tough dataset design. This is an exhaustive survey, but the authors acknowledge that there may be methods that have escaped their attention. They also caution that the analysis provided, is but qualitative. Finally, we mentioned few avenues of further exploration such as the use of semantically rich models, informed dataset design and incorporation of domain knowledge.

References

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, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Yefim Bakman. 2007. Robust understanding of word problems with extraneous information. *arXiv preprint math/0701393*.

Daniel G Bobrow. 1964. A question-answering system for high school algebra word problems. In *Proceedings of the October 27-29, 1964, fall joint computer conference, part I*, pages 591–614. ACM.

Yixuan Cao, Feng Hong, Hongwei Li, and Ping Luo. 2021. A bottom-up dag structure extraction model for math word problems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):39–46.

Oishik Chatterjee, Aashish Waikar, Vishwajeet Kumar, Ganesh Ramakrishnan, and Kavi Arya. 2021. A weakly supervised model for solving math word problems. *CoRR*, abs/2104.06722.

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

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, Volume 1 (Long and Short Papers)*, pages 2656–2668, Minneapolis, Minnesota. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, 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, Doha, Qatar. Association for Computational Linguistics.

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

Denise Dellarosa. 1986. A computer simulation of children's arithmetic word-problem solving. *Behavior Research Methods, Instruments, & Computers*, 18(2):147–154.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Anton Dries, Angelika Kimmig, Jesse Davis, Vaishak Belle, and Luc De Raedt. 2017. Solving probability problems in natural language. *Proceedings Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 3981–3987.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Charles R Fletcher. 1985. Understanding and solving arithmetic word problems: A computer simulation. *Behavior Research Methods, Instruments, & Computers*, 17(5):565–571.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Kaden Griffith and Jugal Kalita. 2020. Solving arithmetic word problems using transformer and preprocessing of problem texts. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 76–84, Indian Institute of Technology Patna, Patna, India. NLP Association of India (NLPAI).

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

Yining Hong, Qing Li, Daniel Ciao, and Song-Chun Zhu. 2021. Learning by fixing: Solving math word problems with weak supervision.

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)*, pages 523–533.

Danqing Huang, Shuming Shi, Chin-Yew Lin, and Jian Yin. 2017. Learning fine-grained expressions to solve math word problems. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 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 (Volume 1: Long Papers)*, pages 887–896, Berlin, Germany. Association for Computational Linguistics.

Danqing Huang, Jin-Ge Yao, Chin-Yew Lin, Qingyu Zhou, and Jian Yin. 2018. Using intermediate representations to solve math word problems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 419–428, Melbourne, Australia. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2.

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*, 3:585–597.

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.

Vivek Kumar, Rishabh Maheshwary, and Vikram Pudi. 2021. Adversarial examples for evaluating math word problem solvers. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2705–2712, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. 2014. Learning to automatically solve algebra word problems. *ACL (1)*, pages 271–281.

Yihuai Lan, Lei Wang, Qiyuan Zhang, Yunshi Lan, Bing Tian Dai, Yan Wang, Dongxiang Zhang, and Ee-Peng Lim. 2021. Mwptoolkit: An open-source framework for deep learning-based math word problem solvers. *arXiv preprint arXiv:2109.00799*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Phuc H. Le-Khac, Graham Healy, and Alan F. Smeaton. 2020. Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934.

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*, pages 6162–6167, Florence, Italy. Association for Computational Linguistics.

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. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2841–2852, Online. Association for Computational Linguistics.

Zhongli Li, Wenxuan Zhang, Chao Yan, Qingyu Zhou, Chao Li, Hongzhi Liu, and Yunbo Cao. 2021. Seeking patterns, not just memorizing procedures: Contrastive learning for solving math word problems. *CoRR*, abs/2110.08464.

Zhenwen Liang, Jipeng Zhang, Jie Shao, and Xiangliang Zhang. 2021. MWP-BERT: A strong baseline for math word problems. *CoRR*, abs/2107.13435.

Zhenwen Liang and Xiangliang Zhang. 2021. Solving math word problems with teacher supervision. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3522–3528. International Joint Conferences on Artificial Intelligence Organization. Main Track.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):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. *arXiv preprint arXiv:1705.04146*.

Qianying Liu, Wenyu Guan, Sujian Li, Fei Cheng, Daisuke Kawahara, and Sadao Kurohashi. 2022. Roda: Reverse operation based data augmentation for solving math word problems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:1–11.

Qianying Liu, Wenyv Guan, Sujian Li, and Daisuke Kawahara. 2019. 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, Hong Kong, China. Association for Computational Linguistics.

Sandra P. Marshall. 1996. *Schemas in Problem Solving*. Cambridge University Press.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2021. A diverse corpus for evaluating and developing english math word problem solvers. *CoRR*, abs/2106.15772.

Arindam Mitra and Chitta Baral. 2016. Learning to use formulas to solve simple arithmetic problems. ACL.

Anirban Mukherjee and Utpal Garain. 2008. A review of methods for automatic understanding of natural language mathematical problems. *Artificial Intelligence Review*, 29(2):93–122.

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*, pages 2080–2094, Online. Association for Computational Linguistics.

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

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Piotr Piękos, Mateusz Malinowski, and Henryk Michalewski. 2021. Measuring and improving BERT's mathematical abilities by predicting the order of reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 383–394, Online. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 5870–5881, Online. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

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

Subhro Roy and Dan Roth. 2017. Unit dependency graph and its application to arithmetic word problem solving. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3082–3088. AAAI Press.

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

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

Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. 2015. Solving geometry problems: Combining text and diagram interpretation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 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 2021*, pages 2269–2279, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Shuming Shi, Yuehui Wang, Chin-Yew Lin, Xiaojiang Liu, and Yong Rui. 2015. Automatically solving number word problems by semantic parsing and reasoning. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1132–1142.

Sowmya S. Sundaram and Savitha Sam Abraham. 2019. Semantic representation for age word problems with schemas. *New Generation Computing*, 37(4):429–452.

Sowmya S Sundaram, Deepak P, and Savitha Sam Abraham. 2020. Distributed representations for arithmetic word problems. *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Simon Suster, Pieter Fivez, Pietro Totis, Angelika Kimmig, Jesse Davis, Luc de Raedt, and Walter Daelemans. 2021. Mapping probability word problems to executable representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3627–3640, Online and

Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

Shih-hung Tsai, Chao-Chun Liang, Hsin-Min Wang, and Keh-Yih Su. 2021. Sequence to general tree: Knowledge-guided geometry word problem solving. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 964–972, Online. Association for Computational Linguistics.

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: Volume 1, Long Papers*, pages 494–504, Valencia, Spain. Association for Computational Linguistics.

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

Lei Wang, Dongxiang Zhang, Lianli Gao, Jingkuan Song, Long Guo, and Heng Tao Shen. 2018. Mathdqn: Solving arithmetic word problems via deep reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

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*, pages 845–854.

Stephen Wolfram. 2015. Wolframlalpha. *On the WWW. URL http://www. wolframalpha. com.*

Qinzhuo Wu, Qi Zhang, Jinlan Fu, and Xuanjing 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, Online. Association for Computational Linguistics.

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 2021*, pages 1473–1482, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qinzhuo Wu, Qi Zhang, Zhongyu Wei, and Xuanjing 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 (Volume 1: Long Papers)*, pages 5859–5869, Online. Association for Computational Linguistics.

Mengzhou Xia, Guoping Huang, Lemao Liu, and Shuming Shi. 2019. Graph based translation memory for neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7297–7304.

Zhipeng Xie and Shichao Sun. 2019. A goal-driven tree-structured neural model for math word problems. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5299–5305. International Joint Conferences on Artificial Intelligence Organization.

Weijiang Yu, Yingpeng Wen, Fudan Zheng, and Nong Xiao. 2021. Improving math word problems with pre-trained knowledge and hierarchical reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3384–3394, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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. 2020a. 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. 2020b. Teacher-student networks with multiple decoders for solving math word problem. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4011–4017. International Joint Conferences on Artificial Intelligence Organization. Main track.

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

Lipu Zhou, Shuaixiang Dai, and Liwei Chen. 2015. Learn to solve algebra word problems using quadratic programming. In *EMNLP*, pages 817–822.