Practice Makes a Solver Perfect: Data Augmentation for Math Word Problem Solvers

Vivek Kumar, Rishabh Maheshwary and Vikram Pudi
Data Sciences and Analytics Center, Kohli Center on Intelligent Systems
International Institute of Information Technology, Hyderabad, India
{vivek.k, rishabh.maheshwary}@research.iiit.ac.in, vikram@iiit.ac.in

Abstract

Existing Math Word Problem (MWP) solvers have achieved high accuracy on benchmark datasets. However, prior works have shown that such solvers do not generalize well and rely on superficial cues to achieve high performance. In this paper, we first conduct experiments to showcase that this behaviour is mainly associated with the limited size and diversity present in existing MWP datasets. Next, we propose several data augmentation techniques broadly categorized into *Substitution* and *Paraphrasing* based methods. By deploying these methods we increase the size of existing datasets by five folds. Extensive experiments on two benchmark datasets across three state-of-the-art MWP solvers shows that proposed methods increase the generalization and robustness of existing solvers. On average, proposed methods significantly increase the state-of-the-art results by over five percentage points on benchmark datasets. Further, the solvers trained on the augmented dataset performs comparatively better on the challenge test set. We also show the effectiveness of proposed techniques through ablation studies and verify the quality of augmented samples through human evaluation.

Original Problem Problem: Nancy grew 8 potatoes. Sandy grew 5 potatoes. How many potatoes did they grow in total ? True Equation: X = 8+5
Paraphrasing Method Problem: How many potatoes did they grow in all given that nancy grew 8 potatoes and sandy grew 5 potatoes. Equation Label: X = 8+5
Substitution Method Problem: Dwight grew 8 potatoes. Juliette grew 5 potatoes. How many potatoes did they grow together ? Equation Label: X = 8+5

Table 1: A MWP and its augmentation examples generated by our methods with preserved equation labels. Blue and Violet colours denote the changes made after the primary stage and secondary stage respectively.

1 Introduction

A Math Word Problem (MWP) consists of natural language text which describes a world state involving some known and unknown quantities, followed by a question text to determine the unknown values. The task is to parse the problem statement and generate equations that can help find the value of unknown quantities. An example of a simple MWP is shown in Table 1. In recent years, the challenge of solving MWP has gained much attention in the NLP community as it needs the development of commonsense multi step reasoning with numerical quantities. With the rise of deep learning, performance of math solvers has also increased significantly over the years (Wang et al.; Zhang et al.). However, recent analysis conducted in (Kumar et al., 2021) and (Patel et al., 2021) show that these deep learning based solvers rely on shallow heuristics to solve vast majority of problems. They curated adversarial examples and SVAMP challenge set respectively to infer that MWP solvers (1) do not understand the relationship between numbers and their associated entities, (2) do not focus on the question text and (3) ignore word order information. In this paper, we first conduct experiments to establish that the above drawbacks are due to the limited size and diversity of problems present in the existing MWP datasets. Next, we propose various augmentation methods to create diverse and large number of training examples to mitigate these shortcomings. Our methods are focused on: (1) Increasing the number of problems in the existing datasets and (2) enhancing the diversity of the problem set.

Training deep neural models effectively requires large number of data points (Longpre et al., 2020). Constructing large datasets which are annotated, labeled and have MWPs of similar difficulty level is a very expensive and tedious task. To address these key challenges, we resort to data augmenta-

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4194 - 4206
July 10-15, 2022 ©2022 Association for Computational Linguistics

tion techniques. Our motivation behind generating augmentations is that humans require sufficient practice to understand MWPs. Humans learn to solve MWPs by going through a variety of similar examples and slowly become capable enough to tackle variations of similar difficulty levels. We aim to generate augmentations such that sufficient linguistic variations of a similar problem are present in the dataset. These variations will make the solver more robust in tackling MWP, increase their reasoning ability and numerical understanding.

Data augmentation for MWPs is a challenging task as we need to preserve the equation labels while generating new samples (Kumar et al., 2021). The generated samples should be (1) semantically similar to their original counterpart, (2) must have the same numerical values and preserve relationship with their respective entities and (3) should maintain the same sequence of events in the problem text. Existing augmentation methods (Wei and Zou) cannot be directly applied due to the above mentioned reasons. Our methods can be broadly classified as follows:

- **Paraphrasing Methods:** It generates variations of the question text by re-statement such that the semantic and syntactic meaning along with the equation labels is preserved.

- **Substitution Methods:** These methods generate variations of the problem statement by identifying and substituting some of the keywords such that the augmentations are semantically and syntactically correct.

To ensure high quality augmentations [1], we propose a selection algorithm which selects samples that have high similarity with original problem and incur high loss values when tested on existing solvers. This algorithm helps selecting only those samples that can make existing solvers more robust. Further, we also verify the validity and the quality of generated augmentations through human evaluation.

Most of the existing MWP datasets are either in languages other than English or contain problems of varying difficulty levels (Koncel-Kedziorski et al., 2016; Wang et al.; Huang et al., 2016; Amini et al., 2019; Miao et al., 2020). We focus on strengthening existing English language datasets which can facilitate the development of better MWP solvers. We consider datasets containing MWP that can be solved using linear equations in one variable. These datasets include MaWPS (Koncel-Kedziorski et al., 2016) and ASDiv-A (Miao et al., 2020) both having $2,373$ and $1,213$ problems respectively. Following are the key contributions made in this paper:

- To the best of our knowledge, this is the first work that extensively evaluates data augmentation techniques for MWP solving. This is the first attempt to generate MWP problems automatically without manual intervention.

- Accuracy of the state of the art solvers increases after training on the proposed augmented dataset. This demonstrates the effectiveness of our methods. To verify the validity of generated augmentations we conduct human evaluation studies.

- We increase the diversity of the training dataset through augmentations and obtain comparatively better results than state-of-the-art solvers on the SVAMP challenge set.

2 Related Work

Math Word Solvers: Many research efforts have been undertaken in the recent past to solve the challenging MWP task. Broadly, these solvers can be categorized into statistical learning based and deep learning based models. Traditional approaches focused more on statistical machine learning (Kushman et al., 2014; Hosseini et al., 2014) with the aim of categorizing equations into templates and extracting key patterns in the problem text. Recently, due to the advent of deep learning in NLP, solvers have witnessed a considerable increase in their performances. (Wang et al.) modelled MWP task as a sequence to sequence task and used LSTM's (Hochreiter and Schmidhuber, 1997) for learning problem representations. (Chiang and Chen, 2018) focused on learning representations for operators and operands.(Wang et al., 2019; Xie and Sun, 2019) used tree structures for decoding process. (Zhang et al.) modelled question as a graph to map quantities and their attributes. Existing datasets which have been used as benchmark for english language includes MaWPS (Koncel-Kedziorski et al., 2016) and Chinese language dataset Math23K (Wang et al.). These datasets although constrained by

[1] Codebase and augmented datasets are available at: https://github.com/kevivk/MWP-Augmentation

their size deal with algebraic problems of similar difficulty levels. Recently, ASDiv (Miao et al., 2020) has been proposed, which has diverse problems which includes annotations for equations, problem type and grade level. Other large datasets in English language include MathQA (Amini et al., 2019) and Dolphin18k (Huang et al., 2016). Although, these datasets have larger problem set but they are noisy and contain problems of varied difficulty levels.

Text Data Augmentation: To effectively train deep learning models, large datasets are required. Data augmentation is a machine learning technique that artificially enlarges the amount of training data by means of label preserving transformations (Taylor and Nitschke, 2018). (Longpre et al., 2020) hypothesize that textual data augmentation would only be helpful if the generated data contains new linguistic patterns that are relevant to the task and have not been seen in pre-training. In NLP, many techniques have been used for generating augmentations, (Wei and Zou) introduced noise injection, deletion, insertion and swapping of words in text. (Rizos et al., 2019) used recurrent neural networks and generative adversarial networks for short-text augmentation (Maheshwary et al., 2021b). Recently, hard label adversarial attack models have also been used (Maheshwary et al., 2021a). Other frequently used methods include inducing spelling mistakes (Belinkov and Bisk, 2018), synonym replacement (Zhang et al., 2016), identifying close embeddings from a defined search space (Alzantot et al., 2018), round trip translations (Sennrich et al., 2016), paraphrasing techniques (Kumar et al., 2019) and words predicted by language model (Kobayashi, 2018) among many others. These methods are specific to the task at hand and needs to be adapted such that the generated augmentations bring diversity in the concerned dataset.

3 Proposed Augmentation Approach

Data augmentation generates new data by modifying existing data points through transformations based on prior knowledge about the problem domain. We introduce carefully selected transformations on well known text augmentation techniques to develop examples suited for the task of MWP. These transformations help in increasing the diversity and size of problem set in existing datasets.

3.1 Problem Definition

A MWP is defined as an input of n tokens, $\mathcal{P} = \{w_1, w_2..w_n\}$ where each token w_i is either a numeric value or a word from a natural language. The goal is to generate a valid mathematical equation $\mathcal{E_P}$ from $\mathcal{P}$ such that the equation consists of numbers from $\mathcal{P}$, desired numerical constants and mathematical operators from the set $\{/, *, +, -, =, (,)\}$. Let $\mathcal{F} : \mathcal{P} \rightarrow \mathcal{E_P}$ be an MWP solver where $\mathcal{E_P}$ is the equation to problem $\mathcal{P}$. Our task is to generate augmented problem statement $\mathcal{P}^*$ from the original input $\mathcal{P}$ such that $\mathcal{P}^*$ is: (1) semantically similar to the initial input $\mathcal{P}$, (2) preserves the sequence of events in the problem statement, (3) keeps the numerical values intact and (4) the solution equation is same as $\mathcal{E_P}$.

3.2 Deficiencies in Existing Models

As showcased by (Patel et al., 2021), existing MWP solvers trained on benchmark datasets like MaWPS and ASDiv-A focus their attention only on certain keywords in the problem statement and do not pay much heed to the question text. We further show that even after performing significant transformations on the test set such as (1) dropping the question text, (2) randomly shuffling the sequence of sentences, (3) random word deletion, and (4) random word reordering, the solvers are still able to produce correct equations. Upon introducing these transformations we should expect a very high drop in accuracy values as the transformed problems are now distorted. Surprisingly, the decrease in accuracy scores is relatively very less than expected as shown in Table 2. We only observe a relatively moderate drop for word reordering. From this analysis, we can say that instead of focusing on the sequence of events, question text and semantic representation of the problem, solvers pick word patterns and keywords from the problem statement. We hypothesize that the drop in accuracy for word reordering experiment indicates that the solvers try to identify a contiguous window of words having some keywords and numbers in them, and generates equation based on these keywords. We further probe on this hypothesis by visualizing the attention weights in the experiment section.

3.3 Augmentation Methods

A MWP can also be expressed as $\mathcal{P} = (S_1, S_2..S_k, Q)$ where Q is the question and $(S_1, S_2..S_k)$ are the sentences constituting the prob-

Dataset	Eval Type	Seq2Seq	GTS	Graph2Tree
MaWPS	True	84.6	87.5	88.7
	WD	**80.2**	**81.5**	**77.3**
	QR	**77.4**	**82.0**	**80.2**
	SS	**77.0**	**60.4**	**66.4**
	WR	**54.9**	**34.8**	**39.3**
ASDiv-A	True	70.6	80.3	82.7
	WD	**60.2**	**61.3**	**56.7**
	QR	**58.7**	**52.4**	**54.1**
	SS	**56.2**	**59.3**	**60.7**
	WR	**47.1**	**32.3**	**34.6**

Table 2: Performance of solvers on modified test sets. True represents unaugmented test set. WD,QR,SS,WR represent word deletion, question reordering, sentence shuffling and word reordering respectively.

lem description. To mitigate the deficiencies in MWP solvers, we propose a two stage augmentation paradigm consisting of primary and secondary stage. In primary stage, we generate base augmentation candidates which then proceed to the secondary stage and get modified accordingly to become potential candidates. After identifying the potential candidates, we filter out the best candidates using proposed candidate selection algorithm. Table 1 shows changes in MWP after primary and secondary stage. Following are the details:

- **Primary Stage:** In the primary stage, our focus is on inducing variations in the question text Q of a given problem statement $\mathcal{P}$. For this, we first generate n base candidates $\{b_1, b_2, ..., b_n\}$ from Q using $T5$ paraphrasing model (Raffel et al., 2020). The key intuition behind this step is to ensure that each augmentation of a given problem has a different question text. This will empower the solver to learn variations from the question text as well.

- **Secondary Stage:** After the generation of base candidates, we implement augmentation methods to generate potential candidates. These methods although well known, require careful tuning to adapt for MWP generation. Table 3 showcases MWP examples and their generated augmentations. Detailed description of these techniques follow.

3.3.1 Paraphrasing Methods

Paraphrasing has proved to be an effective way of generating text augmentations (Witteveen and Andrews). It generates samples having diverse sentence structures and word choices while preserving the semantic meaning of the text. These additional samples guide the model to pay attention to not only the keywords but its surroundings as well. This is particularly beneficial for the task of MWP solving, where most of the problem statements follow a general structure.

Problem Reordering: Given original problem statement $\mathcal{P} = (S_1, S_2, ...S_k, Q)$, we alter the order of problem statement such that $\mathcal{P}^* = (Q, S_1, S_2, ..., S_k)$. To preserve the semantic and syntactic meaning of problem statement we use filler phrases like 'Given that' and 'If-then'. To make these paraphrases more fluent, we use named entity recognition and co-reference resolution to replace the occurrences of pronouns with their corresponding references. Please note that this method is better than random shuffling of sentences as it preserves the sequence of events in the problem statement.

Round Trip Translations: Round trip translations, more commonly referred as back-translation is an interesting method to generate paraphrases. This idea has evolved as a result of the success of machine translation models (Wu et al., 2016). In this technique, sentences are translated from their original language to foreign languages and then translated back to the original language. This round trip can be between multiple languages as well. The motivation behind using this technique is to utilize the different structural constructs and linguistic variations present in other languages.
Back-translation is known to diverge uncontrollably (Tan et al., 2019) for multiple round trips. This may lead to change in the semantics of the problem statement. Numerical quantities are fragile to translations and their order and representation may change. To overcome these challenges, we worked with languages that have structural constructs similar with English. For instance, languages like Finnish which are gender neutral, can become problematic as they can lead to semantic variance in augmented examples. To preserve numerical quantities, we replace them with special symbols and keep a map to restore numerical quan-

Category	Augmentation Method	Example
Paraphrasing Methods	**Round trip Translation**	**Original:** The schools debate team had 4 boys and 6 girls on it. If they were split into groups of 2, how many groups could they make ? **Augmented:** The school discussion group consisted of 4 boys and 6 girls. If they are divided into groups of 2 . How many groups could they have created ?
	Problem Reordering	**Original:** Lucy has an aquarium with 5 fish . She wants to buy 1 more fish . How many fish would Lucy have then ? **Augmented:** If lucy has an aquarium with 5 fish and she wants to buy 1 more fish then how many fish would lucy have ?
Substitution Methods	**Fill Masking**	**Original:** There are 8 walnut trees currently in the park . Park workers will plant 3 more walnut trees today . How many walnut trees will the park have when the workers are finished ? **Augmented:** There are 8 walnut trees currently in the park . Park workers will plant 3 more walnut trees soon . How many walnut trees will the park have after the workers are finished ?
	Named-Entity Replacement	**Original:** Sally found 7 seashells , Tom found 12 seashells , and Jessica found 5 seashells on the beach . How many seashells did they find together ? **Augmented:** Edd found 7 seashells , Alan found 12 seashells , and Royal found 5 seashells on the beach . How many seashells were found together ?
	Synonym Replacement	**Original:** Katie 's team won their dodgeball game and scored 25 points total . If Katie scored 13 of the points and everyone else scored 4 points each , how many players were on her team ? **Augmented:** Katie's group won their rumble game and scored 25 points total . If Katie scored 13 of the points and all else scored 4 points each, How many players was on her group ?

Table 3: Augmentation examples from all proposed methods. Coloured text represents the changes in problem statement.

tities in the generated paraphrases. We have used the following round trips:

English - Russian - English: Although Russian is linguistically different from English, we still chose it as word order does not affect the syntactic structure of a sentence in Russian language (Voita et al., 2019). For single round trip, we preferred Russian as it has the potential to generate different paraphrase structures.

English - German - French - English: German and french are structurally similar to English language (Kim et al., 2019), we chose them for multiple round trips to both maintain semantic in-variance and induce minor alterations in the paraphrases.

3.3.2 Substitution Methods

In this class of methods, the focus is on generating variations of the problem statement by identifying and substituting some of the keywords such that the augmentations are semantically and syntactically correct, with the equation labels preserved. Substitution is effective for MWP solving as it guides the solvers focus away from certain keywords, allowing it to distribute its attention and generalize better. We propose the following methods:

Fill-Masking: In this technique, we model the challenge of generating candidates as a masked language modelling problem. Instead of randomly choosing words for masking, we use part of speech tags to focus on nouns and adjectives, preferably in the vicinity of numerical quantities. We replace these identified keywords with mask tokens. These masked candidate sentences are then passed through a masked language model (Devlin et al., 2019a) and suitable words are filled in masked positions to generate our candidate sentences.

Synonym Replacement: In this method, after stop-word removal, we select keywords randomly for substitution. Unlike fill-mask technique, where masked language models were deployed, here we use Glove embeddings (Pennington et al., 2014) to find the top k candidates that are close synonyms of the keywords. To ensure syntactic correctness in candidates, we maintain the part of speech tags for the substitute candidates. These synonyms are then used to substitute the keywords in the problem statement and generate augmented candidates.

Named-Entity Replacement: A common occurrence in MWP is the usage of named entities. These entities play a crucial role in stating the problem statement, but the solution equations do not change on altering these entities. Following this insight, we first identify the named entities[2] such as person, place and organizations present in the problem statement. Then we replace these named entities with their corresponding substitutes, like a person's name is replaced by another person's name to generate the potential candidates. Table 4 reports the statistics of augmented datasets on both MaWPS and ASDiv-A. All the techniques described in paraphrasing and substitution methods are used for generating the potential candidates for a problem statement. After generation of the potential candidates for augmenting a problem statement, the best possible candidate is selected by using Algorithm 1. Key motivation behind developing this algorithm is to select candidates on which the solver does not perform well and which are similar to the original problem statement.

We use negative log likelihood as the loss function $\mathcal{L}$ and Sentence-BERT (Reimers and Gurevych, 2019) fine tuned on MWP equation generation task as sentence embedding generator $\mathcal{S}$. We calculate the similarity of each candidate embedding with the original problem representation using cosine similarity as shown in Line 3 of the algorithm. Further, for each candidate sentence, we evaluate their loss values and select the candidate with the maximum mean normalized loss and similarity score.

Dataset	Problem Size	Vocabulary Size
MaWPS	2,373	2,632
ASDiv-A	1,213	2,893
Paraphrase	5,909	3,832
Substitution	6,647	3,923
Combined-MaWPS	**10,634**	**5,626**
Combined-ASDiv	**5,312**	**6,109**

Table 4: Statistics of augmented dataset compared with MaWPS and ASDiv-A. Combined-Dataset represents combination of Paraphrase and Substitution methods.

4 Experiments

Datasets and Models: To showcase the effectiveness of proposed augmentation methods, we select three state-of-the-art MWP solvers: (1) *Seq2Seq* (Wang et al.) having an LSTM encoder and an attention based decoder. (2) *GTS* (Xie and Sun, 2019) having an LSTM encoder and a tree based decoder and (3) *Graph2tree* (Zhang et al.) consists of a both tree based encoder and decoder. *Seq2Seq* serves as our base model for experimentation. Many existing datasets are not suitable for our analysis as either they are in Chinese (Wang et al.) or they have problems of higher complexities (Huang et al., 2016) . We conduct experiments across the two largest available English language datasets satisfying our requirements: (1) *MaWPS* (Koncel-Kedziorski et al., 2016) containing $2,373$ problems (2) *ASDiv-A* (Miao et al., 2020) containing $1,213$ problems. Both datasets have MWPs with linear equation in one variable.

[2]https://www.nltk.org/

Algorithm 1 MWP Candidate Selection Algorithm

Requires: $\mathcal{M}$ is augmentation method, $\mathcal{S}$ is similarity model, $\mathcal{F}$ is solver model, $\mathcal{L}$ is Loss function.
Input: Problem text $\mathcal{P}$
Output: Augmented Text $\mathcal{P}^*$

1: $\mathcal{E}_P \leftarrow \mathcal{F}(\mathcal{P})$
2: $Candidates \leftarrow \mathcal{M}(\mathcal{P})$
3: **for** C_j in $Candidates$: **do**
4: $\quad S_j \leftarrow \mathcal{S}(C_j, \mathcal{P})$
5: $\quad L_j \leftarrow (\mathcal{L}(C_j) - \mathcal{L}(P))/\mathcal{L}(P)$
6: $\quad CandidateScore.add(S_j * L_j)$
7: $\mathcal{P}^* = \arg\max_{C_j} CandidateScore(C_j)$
8: **end**

Experiment Setup: We train and evaluate the three solvers on both MaWPS and ASDiv-A using five fold cross validation. Evaluation is conducted on both original and augmented datasets. We use the same hyperparameter values as recommended in the original implementation of these solvers. Further, each solver has been trained from scratch and by using BERT embeddings (Devlin et al., 2019b). We also evaluate the models on *SVAMP* (Patel et al., 2021) challenge set. This test set has been designed specifically to examine the robustness and adaptability of the solvers. Ablation studies have been conducted to assess the effectiveness of candidate selection algorithm and augmentation techniques.

Dataset	Evaluation Type	Seq2Seq	GTS	G2T
MaWPS	True	84.6	87.5	88.7
	Paraphrasing	**88.3**	**90.4**	**92.6**
	Substitution	**89.2**	**89.7**	**91.7**
	Combined	**91.3**	**92.6**	**93.5**
ASDiv-A	True	70.6	80.3	82.7
	Paraphrasing	**75.6**	**84.2**	**83.6**
	Substitution	**73.2**	**83.3**	**84.1**
	Combined	**78.2**	**85.9**	**86.3**

Table 5: Result of augmentation methods. True is original dataset, Combined is combination of paraphrasing and substitution. G2T represents Graph2Tree solver.

Problem 1: Ricardo was making baggies of cookies with 5 cookies in each bag. If he had 7 chocolate chip cookies and 3 oatmeal cookies, how many baggies could he make ?
Solution Equation: X = (7+3)/5
Pre Augmentation Equation: X = (7/3)/3
Post Augmentation Equation: X = (7+3)/5

Problem 2: For halloween Destiny bought 9 pieces of candy. She ate 3 pieces the first night and then her sister gave her 2 more pieces. How many pieces of candy does Destiny have now ?
Solution Equation: X = 9-3+2
Pre Augmentation Equation: X = ((9+3-3
Post Augmentation Equation: X = (9+3-2)

Problem 3 : Audrey needs 6 cartons of berries to make a berry cobbler. She already has 2 cartons of strawberries and 3 cartons of blueberries. How many more cartons of berries should Audrey buy ?
Solution Equation: X = 6-2-3
Pre Augmentation Equation: X = (6-(2)+3)
Post Augmentation Equation: X = 6-(2+3)

Table 6: Examples illustrating equation results before and after training on the full augmented dataset.

4.1 Results and Analysis

Table 5 shows the result of proposed methods. These results have been reported on BERT embeddings. Table 11 shows a comparison between training from scratch and using BERT embeddings. By training these state-of-the-art models on the augmented dataset we achieve better results for both MaWPS and ASDiv-A. On average, we were able to increase the accuracy significantly by more than five percentage points. Both paraphrasing and substitution methods have performed well independently and in combination. Further, we conduct ablation studies to analyze the performance of each augmentation method. In Table 6 we illustrate some examples on which existing models generate incorrect equations. However, after being trained with augmented dataset they generate correct equations. Additionally, in Problem 2 the base model generates syntactically incorrect solution, but post augmentation it generates syntactically correct equation. These examples show the increased robustness and solving abilities of solvers.

Problem: Gavin has 6 shirts . 3 are blue the rest are green. How many green shirts does Gavin have ?
Mean attention values: 0.27 0.14 0.08

Problem: Gavin has 6 shirts . 3 are blue the rest are green. How many green shirts does Gavin have ?
Augmented mean attention values : 0.23 0.18 0.11

Problem: There are 3 pencils in the drawer. Sara placed 7 more pencils in the drawer. How many pencils are there in all ?
Mean attention values: 0.45 0.11 0.05

Problem: There are 3 pencils in the drawer. Sara placed 7 more pencils in the drawer. How many pencils are there in all ?
Augmented mean attention values : 0.31 0.16 0.09

Table 7: Examples illustrating distribution of top three attention weights before and after training on the full augmented dataset.

Attention Visualizations: Through this investigation, we aim to ascertain our hypothesis that to generate equations MWP solvers focus only on certain keywords and patterns in a region. They ignore essential information like semantics, sequence of events and content of the question text present in the problem statement. In Table 7, we show some sample problem statements with their attention weights. These weights are generated during the decoding process using Luong attention mechanism (Luong et al., 2015). Moreover, to illustrate the effectiveness of our augmentation techniques, we show the distribution of attention weights for models trained on the augmented dataset. We can infer from the examples showcased in Table 7 that before augmentation the focus of the solver is limited to a fixed region around numerical quantities and it does not pay heed to the question text. However, after training on the augmented dataset the solver has a better distribution of attention weights, the weights are not localised and and the model is also able to pay attention on the question text.

Ablation Studies: To assert the effectiveness of our methods, we conduct the following ablations:

Method	Eval Type	Seq2Seq	GTS	Graph2Tree
RSA	True	84.6	87.5	88.7
	Paraphrasing	**85.3**	**88.1**	**89.2**
	Substitution	**86.8**	**87.3**	**87.9**
	Combined	**87.0**	**89.2**	**89.5**
CSA	True	84.6	87.5	88.7
	Paraphrasing	**88.3**	**90.4**	**92.6**
	Substitution	**89.2**	**89.7**	**91.7**
	Combined	**91.3**	**92.6**	**93.5**

Table 8: Ablation Study for Random Selection Algorithm (RSA) and Candidate Selection Algorithm (CSA).

Candidate Selection Algorithm: For testing the usefulness of candidate selection algorithm, we compare it with a random selection algorithm. In this, we randomly select one of the possible candidates as augmented problem statement. We evaluate the accuracy of models trained on the augmented datasets, generated using both the algorithms. Result in Table 8, shows that candidate selection algorithm performs better than random selection algorithm and this demonstrates the effectiveness of our algorithm.

Augmentation Methods: To examine the effectiveness of proposed augmentation techniques, we evaluate the models on each of the proposed techniques independently and report the results in Table 9. Although, all the methods contribute towards increase in accuracy but Round trip translations and synonym replacement perform marginally better than others. This behaviour can be linked to the structural diversity and keyword sensitivity that round trip translations and synonym replacement bring respectively (Feng et al., 2021).

Augmentation	Seq2Seq	GTS	Graph2Tree
True	84.6	87.5	88.7
RRT	**86.5**	89.1	**91.6**
PR	85.9	88.4	90.7
FM	84.8	87.2	89.1
SR	85.2	**90.1**	91.2
NER	86.1	88.3	89.7

Table 9: Result of Ablation study for each augmentation method. True represents unaugmented MaWPS dataset, RRT, PR, FM, SR, NER represents round trip translations, problem reordering, fill masking,synonym replacement and named entity replacement respectively.

SVAMP Challenge Set: SVAMP (Patel et al., 2021) is a manually curated challenge test set consisting of $1,000$ math word problems. These problems have been cherry picked from MaWPS and ASDiv-A, then altered manually to modify the semantics of question text and generate additional equation templates. This challenge set is suitable for evaluating a solver's performance as it modifies problem statements such that solver's generalization can be checked. The results are shown in Table 10. Although, our proposed augmented dataset has very limited equation templates, still it performs comparatively better than state-of-the-art models on SVAMP challenge set. This result signifies the need for a larger and diverse dataset with enhanced variety of problems. Further, it demonstrates the effectiveness of our method which is able to perform better on SVAMP test set and increase model's accuracy despite the challenges.

Augmentation	Seq2Seq	GTS	Graph2Tree
True	37.5	39.6	41.2
MaWPS(P+S)	39.2	40.1	42.3
ASDiv-A(P+S)	37.8	40.4	42.1
Combined	**40.2**	**41.3**	**43.8**

Table 10: Result of augmentations on SVAMP Challenge Set. P and S represent paraphrasing and substitution methods. Combined represents augmented MaWPS and ASDiv-A. True is combined MaWPS and ASDiv-A.

Augmentation Method	MaWPS		ASDiv-A	
	Scratch	BERT	Scratch	BERT
True	77.2	**84.6**	53.2	**70.6**
Paraphrasing	79.8	**88.3**	58.1	**75.6**
Substitution	81.3	**89.2**	57.3	**73.2**
Combined	82.7	**91.3**	60.4	**78.2**

Table 11: Performance comparison of baseline model trained from scratch and trained using BERT embeddings. True represents unaugmented dataset.

BERT Embeddings: We train the solvers in two different settings, using pre-trained BERT embeddings and training from scratch. We chose BERT specifically as we require contextual embeddings which could be easily adapted for the task of MWP. Moreover, existing models have also shown results using BERT and it would be fair to compare their performances when trained using similar embeddings. Results obtained are shown in Table 11. We observe that for solver's trained using BERT, accuracy is higher than models trained from scratch.

Human Evaluation: To verify the quality of augmented examples, we conduct human evaluation. The focus of this evaluation is: (1) To check if the augmentations will result in the same linear equation as present in the original problem statement, (2) To evaluate if the numerical values for each augmentation example is preserved, (3) Evaluate each sample in the range 0 to 1 for its semantic similarity with the original problem statement, (4) On a scale of 1 to 5 rate each augmented example for its grammatical correctness. We conduct the human evaluations on randomly shuffled subsets consisting of around 40% of the total augmented examples for both the datasets. This process is repeated three times with different subsets, five human evaluators evaluate each example in all subsets, and the mean results are computed as shown in Table 12.

Evaluation Criteria	**MaWPS**		**ASDiv-A**	
	Para	Sub	Para	Sub
Preserves Equation	92.3%	89.5%	93.6%	90.1%
Preserves Numbers	88.4%	91.2%	87.3%	90.3%
Semantic Similarity	0.96	0.89	0.91	0.87
Syntactic Similarity	4.67	4.36	4.59	4.33

Table 12: Human Evaluation scores on augmentated dataset. Para and Sub represents paraphrasing and substitution methods respectively.

5 Future Work and Conclusion

We showcase that the existing MWP solvers are not robust and do not generalize well on even simple variations of the problem statement. In this work, we have introduced data augmentation techniques for generation of diverse math word problems. We were able to enhance the size of existing dataset by 5 folds and significantly increase the performance of state-of-the-art solvers by over 5 percentage points. Future works could focus on developing techniques to generate data artificially and making robust MWP solvers.

6 Acknowledgment

We thank IHub-Data, IIIT Hyderabad[3] for financial support.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples.

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.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation.

Ting-Rui Chiang and Yun-Nung Chen. 2018. Semantically-aligned equation generation for solving and reasoning math word problems. *arXiv preprint arXiv:1811.00720*.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. Bert: Pre-training of deep bidirectional transformers for language understanding.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp.

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

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. pages 523–533, Doha, Qatar. Association for Computational Linguistics.

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.

Yunsu Kim, Petre Petrov, Pavel Petrushkov, Shahram Khadivi, and Hermann Ney. 2019. Pivot-based transfer learning for neural machine translation between non-English languages. pages 866–876, Hong Kong, China. Association for Computational Linguistics.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations.

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*

[3]https://ihub-data.iiit.ac.in/

of the Association for Computational Linguistics: Human Language Technologies, pages 1152–1157, San Diego, California. Association for Computational Linguistics.

Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha Talukdar. 2019. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. pages 3609–3619, Minneapolis, Minnesota. Association for Computational Linguistics.

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. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 271–281, Baltimore, Maryland. Association for Computational Linguistics.

Shayne Longpre, Yu Wang, and Christopher DuBois. 2020. How effective is task-agnostic data augmentation for pretrained transformers?

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation.

Rishabh Maheshwary, Saket Maheshwary, and Vikram Pudi. 2021a. Generating natural language attacks in a hard label black box setting. In *AAAI*.

Rishabh Maheshwary, Saket Maheshwary, and Vikram Pudi. 2021b. A strong baseline for query efficient attacks in a black box setting.

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

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems?

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Georgios Rizos, K. Hemker, and Björn Schuller. 2019. Augment to prevent: Short-text data augmentation in deep learning for hate-speech classification. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data.

Hao Tan, Licheng Yu, and Mohit Bansal. 2019. Learning to navigate unseen environments: Back translation with environmental dropout.

Luke Taylor and Geoff Nitschke. 2018. Improving deep learning with generic data augmentation. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1542–1547.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. Context-aware monolingual repair for neural machine translation.

Lei Wang, D. Zhang, Jipeng Zhang, Xing Xu, L. Gao, B. Dai, and H. Shen. 2019. Template-based math word problem solvers with recursive neural networks. In *AAAI*.

Yan Wang, Xiaojiang Liu, and Shuming Shi. Deep neural solver for math word problems.

Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks.

Sam Witteveen and Martin Andrews. Paraphrasing with large language models.

Yonghui Wu, Mike Schuster, Z. Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason R. Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144.

Zhipeng Xie and Shichao Sun. 2019. A goal-driven tree-structured neural model for math word problems. In *IJCAI*, pages 5299–5305.

Jipeng Zhang, Lei Wang, Roy Ka-Wei Lee, Yi Bin, Yan Wang, Jie Shao, and Ee-Peng Lim. Graph-to-tree learning for solving math word problems.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2016. Character-level convolutional networks for text classification.

7 Appendix

7.1 Implementation Details

For conducting our experiments we have used two Boston SYS-7048GR-TR nodes equipped with NVIDIA GeForce GTX 1080 Ti computational GPU's having 11GB of GDDR5X RAM. All implementations of training and testing is coded in Python with Pytorch framework. The number of parameters range from 20M to 130M for different models. We use negative log likelihood as the loss criterion. Hyper-parameter values were not modified, and we follow the recommendations of the respective models. To reduce carbon footprint from our experiments, we run the models only on a single fold for searching hyperparameter values. We chose the number of base candidates after primary stage n as 7. Generating augmentation examples using Paraphrasing Methods took around 12 minutes on average for MaWPS and 8 minutes for ASDiv-A datasets. Substitution methods took around 5 minutes on average for both MaWPS and ASDiv-A dataset. The experiments conducted by us are not computation heavy. Each of the state-of-the-art models get trained within 5 hrs of time, with Graph2Tree taking the maximum time.

7.2 Additional Augmented Examples

In this section, we present some additional valid as well as invalid augmented examples. Additionally, we also show some more examples with their attention weights. Table 13 shows some additional examples with their attention weight distribution. These weights have been shown for the base model trained before augmentation and after augmentation on MaWPS dataset. Table 14 illustrates some additional problem statements for all the techniques in paraphrasing methods and substitution methods. In Table 15, we present some invalid augmented examples which do not satisfy our human evaluation criteria. These examples are such that they alter the semantics of the original problem statement.

Problem: A magician was selling magic card decks for 2 dollars each . If he started with 25 decks and by the end of the day he had 4 left, how much money did he earn ? Mean attention values: 0.34 0.11 0.09 Problem: A magician was selling magic card decks for 2 dollars each. If he started with 25 decks and by the end of the day he had 4 left, how much money did he earn ? Augmented mean attention values : 0.19 0.18 0.15
Problem: There are 18 pencils in the drawer and 6 pencils on the desk. Dan placed 4 pencils on the desk. How many pencils are there in total ? Mean attention values: 0.21 0.16 0.06 Problem: There are 18 pencils in the drawer and 6 pencils on the desk. Dan placed 4 pencils on the desk. How many pencils are there in total ? Augmented mean attention values : 0.29 0.19 0.12
Problem: Dan has 12 violet marbles, he gave Mary 4 of the marbles. How many violet marbles does he now have ? Mean attention values: 0.23 0.21 0.17 Problem: Dan has 12 violet marbles , he gave Mary 4 of the marbles. How many violet marbles does he now have ? Augmented mean attention values : 0.23 0.18 0.11
Problem: Angela has 7 tickets . Annie gives Angela 5 more. How many tickets does Angela have in all ? Mean attention values: 0.30 0.19 0.15 Problem: Angela has 7 tickets . Annie gives Angela 5 more. How many tickets does Angela have in all ? Augmented mean attention values : 0.29 0.21 0.14
Problem: Maria had 5 bottles of water in her fridge. If she drank 1 of them and then bought 2 more, how many bottles would she have ? Mean attention values: 0.48 0.14 0.04 Problem Maria had 5 bottles of water in her fridge. If she drank 1 of them and then bought 2 more , how many bottles would she have ? Augmented mean attention values : 0.23 0.17 0.11

Table 13: Examples illustrating distribution of top three attention weights before and after training on the full augmented dataset.

Category	Augmentation Method	Example
Paraphrasing Methods	**Round trip Translation**	**Original:** Alyssa's dog had puppies. She gave 2 to her friends. She now has 3 puppies. How many puppies did she have to start with ? **Augmented:** Alyssa's dog had puppies . She gave her friends 2 . She now has 3 puppies . How many puppies did she start ?
	Problem Reordering	**Original:** Rachel was organizing her book case making sure each of the shelves had exactly 3 books on it. If she had 4 shelves of mystery books and 2 shelves of picture books , how many books did she have total ? **Augmented:**How many books did she have given that rachel was organizing her book case making sure each of the shelves had exactly 3 books on it and she had 4 shelves of mystery books and 2 shelves of picture books .
Substitution Methods	**Fill Masking**	**Original:** A cell phone company has a total of 1000 customers across the world . If 740 of its customers live in the United States , how many of its customers live in other countries ? **Augmented:** A mobile phone firm has a network of 1000 customers across the world . If 740 of its customers live in the violetUS, How many customers live in other locations?
	Named-Entity Replacement	**Original:** Daniel had some noodles. He gave 20 noodles to William. Now Daniel only has 11 noodles. How many noodles did Daniel have to begin with ? **Augmented:** Matt had some noodles. He gave 20 noodles to Zeal. Now Matt only has 11 noodles. How many noodles did Matt have initially ? Edd found 7 seashells , Alan found 12 seashells , and Royal found 5 seashells on the beach . How many seashells were found together ?
	Synonym Replacement	**Original:** There are 5 rulers in the drawer. Tim took 3 rulers from the drawer. How many rulers are now in the drawer ? **Augmented:** There are 5 consonants in the drawer. Tim went 3 consonants from the drawer. How many other consonants are in the drawer now ?

Table 14: Valid Augmentation examples from all proposed methods. Coloured text represents the changes in problem statement.

Category	Augmentation Method	Example
Paraphrasing Methods	**Round trip Translation**	**Original:** Kimberly went to the store 6 times last month . She buys 9 peanuts each time she goes to the store . How many peanuts did Kimberly buy last month ? **Augmented:** Kimberly travelled to club six times last month. She buys 9 peanuts every time she goes to the club . How many peanuts did Kimberly buy last year ?
	Problem Reordering	**Original:** Fred has 10 blue marbles . Fred has number1 times more blue marbles than Tim . How many blue marbles does Tim have ? **Augmented:** If fred has 10 blue marbles and fred has number1 more blue marbles than Tim then how many blue marbles does tim have ?
Substitution Methods	**Fill Masking**	**Original:** Sarah had 7 homework problems . She finished 2 of them but still had 3 pages of problems to do . If each page has the same number of problems on it , how many problems are on each page ? **Augmented:** Sarah had 7 of them . She had 2 of them but still had 3 more of them to do . If each more has the same number of them on it, How many them are on each more ?
	Named-Entity Replacement	**Original:** Beverly had 10 dimes in his bank . His sister Maria borrowed 2 of his dimes . How many dimes does Beverly have now ? **Augmented:** Silva had 10 dimes in his bank . His sister Jeanie borrowed a pair of his dimes . How many dimes does Jeanie have now ?
	Synonym Replacement	**Original:** Shawn's team won their dodgeball game and scored 25 points total . If Shawn scored 13 of the points and everyone else scored 4 points each , how many players were on his team ? **Augmented:** Shawn's group won their rumble game and scored 25 points total . If Shawn scored 13 of the points and everyone else scored quarter points each, how many people were there ?

Table 15: Invalid Augmentation examples from all proposed methods. Coloured text represents the changes in problem statement.