MAWPS: A Math Word Problem Repository

Rik Koncel-Kedziorski*[1], Subhro Roy*[2], Aida Amini[1],
Nate Kushman[3], and **Hannaneh Hajishirzi[1]**
[1]University of Washington, [2]University of Illinois Urbana Champaign, [3]Microsoft Research
{kedzior, amini91, hannaneh}@uw.edu, sroy9@illinois.edu, nkushman@microsoft.com

Abstract

Recent work across several AI subdisciplines has focused on automatically solving math word problems. In this paper we introduce MAWPS, an online repository of Math Word Problems, to provide a unified testbed to evaluate different algorithms. MAWPS allows for the automatic construction of datasets with particular characteristics, providing tools for tuning the lexical and template overlap of a dataset as well as for filtering ungrammatical problems from web-sourced corpora. The online nature of this repository facilitates easy community contribution. At present, we have amassed 3,320 problems, including the full datasets used in several prominent works.

1 Introduction

Automatically solving math word problems has proved a difficult and interesting challenge for the AI research community (Feigenbaum et al., 1995). Math word problems serve as a testbed for algorithms that build a precise understanding of what is being asserted in text. Consider the following:

> *Rachel bought two coloring books. One had 23 pictures and the other had 32. After one week she had colored 44 of the pictures. How many pictures does she still have to color?*

To solve such a problem, an algorithm must model implicit and explicit quantities in the text and their relationships through mathematical operations.

Many researchers have taken on this challenge to design data-driven approaches to solve different types of math word problems (Hosseini et al., 2014; Kushman et al., 2014; Roy and Roth, 2015; Zhou et al., 2015; Koncel-Kedziorski et al., 2015). Treatments of this task range from template-matching to narrative-building, and methods including integer linear programming, factorization, and more appear in the literature. As a result of the variety of approaches, several datasets have emerged. The variety of math word problems in these datasets is such that researchers have already begun specializing in the types of problems that their systems are able to successfully solve. Additionally, in some datasets one may find significant repetition of common words, a small set of equation templates used again and again, or problem texts which map to highly degraded grammatical structures. Therefore, designing general algorithms that can address different problem types while still being robust to the variations across datasets has remained a challenge.

In response to the burgeoning interest in this task, we present MAWPS (MAth Word ProblemS, pronounced *mops*), a framework for building an online repository of math word problems. Our framework includes a collection of varying types of math word problems, their answers, and equation templates, as well as interfaces which allow researchers to dynamically update and add more problem types. Additionally, in light of the problematic features of the current datasets cited above, our framework provides the possibility for customizing a dataset with regard to considerations such as lexical and template overlap or grammaticality, allowing researchers to choose how many of the difficulties of open domain web-sourced word problem texts they want

* denotes equal contribution

Proceedings of NAACL-HLT 2016, pages 1152–1157,
San Diego, California, June 12-17, 2016.

to tackle. We report the important characteristics of the current available data as well as the performance of some state-of-the-art systems on various subsets of the data. MAWPS is located at `http://lang.ee.washington.edu/MAWPS`.

2 Related Work

Recently, automatically solving math word problems has attracted several researchers. Specific topics include number word problems (Shi et al., 2015), logic puzzle problems (Mitra and Baral, 2015), arithmetic word problems (Hosseini et al., 2014; Roy and Roth, 2015), algebra word problems (Kushman et al., 2014; Zhou et al., 2015; Koncel-Kedziorski et al., 2015), and geometry word problems (Seo et al., 2015).

A few recent works, Kushman et al. (2014) and Zhou et al. (2015) focus on solving math word problems by matching quantities and variables (nouns) extracted from the problem text to templates appearing in the training data. These methods have a broad scope, but they rely heavily on overlap between templates in the training and test data. As shown in our experiments, when that overlap is reduced, the performance of the systems drops significantly.

Other work has focused on more limited domains, but aims to reduce the reliance on data overlap. Hosseini et al. (2014) solve addition and subtraction problems by learning to categorize verbs for the purpose of updating a world representation derived from the problem text. Roy and Roth (2015) treat arithmetic word problem templates as equation trees and introduce a method for learning the least governing node for two text quantities. Koncel-Kedziorski et al. (2015) focus on single equation problems and use typed semantically-rich equation trees where nodes correspond to numbers and an associated type derived from the problem text, and efficiently enumerate the space of these trees using integer linear programming.

Our work is also inspired by the recent work in introducing datasets to evaluate question answering and reading comprehension tasks that require reasoning and entailment. In contrast to Richardson et al. (2013), our work is focused on making a dataset for math word problems to evaluate robustness, scalability, and scope of algorithms in quantitative reasoning. In contrast to Weston et al. (2015), our work has more natural text and a larger vocabulary, does not use synthetic data, and is only focused on math word problems which is an extension of the counting sub-category introduced in that work.

3 Data

We compile a dataset of arithmetic and algebra word problems of varying complexity from different websites. This data extends the published word problem datasets used in Hosseini et al. (2014), Kushman et al. (2014), Koncel-Kedziorski et al. (2015), and Roy and Roth (2015). In addition to this strong foundation, the MAWPS repository provides interfaces for adding new word problems, allowing for its extension and development.

Noting that word problem datasets have varied with regard to noisiness and repetition, we define several data characteristics to capture these properties below. We then allow a user of the repository to select a subset of the data that optimizes these characteristics. The three data characteristics that can be automatically manipulated by MAWPS are:

Lexical Overlap. Lexical overlap describes the reuse of lexemes among problems in a dataset. For example, several problems may describe tickets being sold for a play, with only the number of adult and child tickets changed among them. These problems would have high lexical overlap. Previous work has shown that lexeme reuse in a dataset allows for spurious associations between the problem text and a correct solution (Koncel-Kedziorski et al., 2015). Researchers have sought to reduce this characteristic in their data so as to facilitate the development of more robust methods.

Template Overlap. By *template* we mean an equation where numerical values are replaced by a consistent, ordered set of constant variables. For example, the equations $(12 * 2) + 7 = x$ and $(6 * 15) + 105 = x$ have the same template, $(a * b) + c = x$. As shown in Roy and Roth (2015) and elsewhere, the appearance of a similar template in the training data is a requirement for some systems at test time. More general math word problem solvers have been introduced that reduce or eliminate this reliance. MAWPS provides for the automatic construction of a dataset with minimized template overlap.

Grammaticality. Many math word problems for use in AI research are drawn from user-submitted

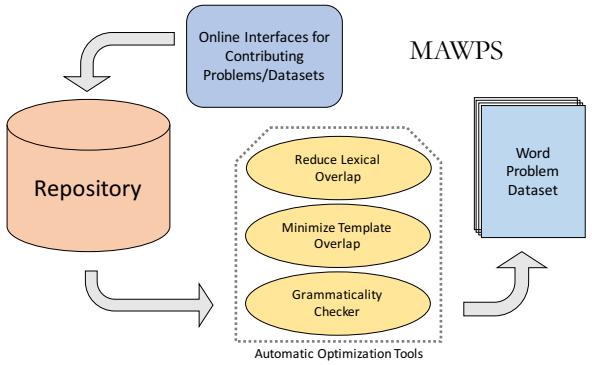

Figure 1: MAWPS System overview: Online interfaces allow for extending repository. Optimization Tools allow for real-time delivery of usable datasets with particular characteristics.

online sources. Some problems from these sources contain grammatical errors including morphological agreement failures, misspellings, and other more complicated ungrammaticalities (for example, "Joan paid $8.77 on a cat toy, and a cage cost her $10.97 with a $20 bill."). While some researchers wish to develop methods robust to these kind of errors, others would focus on the subset of the data that adheres to a high-precision, expertly developed grammar of English. To facilitate both lines of research, we allow for the selection of only those problems which meet this judgment, or the full data.

4 System Overview

The goal of the MAWPS repository is to provide an extendable collection of math word problems which allows researchers to select the portion of the data that meets their needs. To facilitate the growth of the collection, our framework allows for easily adding a single word problem or a whole dataset to the repository. We design backend tools which allow researchers to select a dataset with reduced lexical overlap, minimized template overlap, or improved grammaticality characteristics even as the repository continues to grow through community contribution. Figure 1 shows an outline of the MAWPS system.

4.1 Reduce Lexical Overlap

We consider the lexical overlap of a dataset D to be the average of the pairwise lexical overlap between each pair of problems in D. First, let $W(p)$ denote the set of unique unigrams and bigrams in a problem p. Then $\textit{PairLex}(p,q) = \frac{|W(p)\cap W(q)|}{|W(p)\cup W(q)|}$ denotes the pairwise lexical overlap between problems p and q.

We formally define the lexical overlap of a dataset D as

$$\mathrm{Lex}(D) = \frac{1}{N}\sum_{\substack{p_i,p_j\in D\\ i<j}} \textit{PairLex}(p_i,p_j)$$

where $N = \binom{|D|}{2}$ is the number of problem pairs in D. Given a dataset D and a target subset size k, MAWPS automatically finds a subset D' of reduced lexical overlap by solving the following optimization problem: $\min_{D'\subseteq D,|D'|=k} \mathrm{Lex}(D')$.

This is the remote clique problem[1], which is NP hard. As a result, we resort to a greedy strategy which gives an approximation ratio of 2 (Birnbaum and Goldman, 2007). We define $f_{\mathrm{lex}}(p,D') = \sum_{q\in D'} \textit{PairLex}(p,q)$, which describes the overlap between a problem p and dataset D'. The greedy method to generate subset D' of D is as follows: we iteratively add a problem p from $D\setminus D'$ to D' which minimizes the value of $f_{\mathrm{lex}}(p,D')$. That is, at each step, we add a problem which has the least average pairwise lexical overlap with the problems already in D'. We repeat this process, starting with a randomly initialized singleton set, until a set of k problems is created.

4.2 Minimize Template Overlap

The template overlap of a dataset D is defined analogously to lexical overlap as the average of the pairwise template overlap between each pair of problems of D. Let $\textit{PairTempl}(p,q)$ be 1 if p and q have the same template (corresponding to the gold equation), and 0 otherwise. We then define the template overlap of a dataset D as

$$\mathrm{Tmpl}(D) = \frac{1}{N}\sum_{\substack{p_i,p_j\in D\\ i<j}} \textit{PairTempl}(p_i,p_j)$$

where $N = \binom{|D|}{2}$. Given a dataset D and a target subset size k, the template overlap reduced subset of D is generated by the following: $\min_{D'\subseteq D,|D'|=k} \mathrm{Tmpl}(D')$.

[1]Given a complete graph with nonnegative edge weights satisfying the triangle inequality and a positive integer p, the remote-clique problem is to find a subset of p vertices having a maximum-weight induced subgraph.

Dataset	# Probs$\|D\|$	# Gramm.	Lexical Overlap (Lex)			Template Overlap (Tmpl)		
			$k = \|D\|/2$	$k = \|D\|$	Reduction	$k = \|D\|/2$	$k = \|D\|$	Reduction
AddSub	395	357	6.1	7.9	22.8	33	37.2	11.3
SingleOp	562	491	6.1	7.8	21.8	24.7	25.4	2.8
MultiArith	600	526	7.8	9.4	17.0	19.7	22.1	10.9
SingleEq	508	434	5.4	6.8	20.6	11	17.9	38.5
SimulEq-S	514	437	4.7	6	21.7	2.9	12.5	76.8
SimulEq-L	1155	980	4.4	5.7	22.8	0.1	3.3	97.0

Table 1: Characteristics of datasets added to our repository. # Gramm. is no. of problems which passed our grammaticality check, the Lex column reports minimum lexical overlap for size k dataset, the Tmpl column reports minimum template overlap for size k dataset, Reduction denotes the % decrease in overlap value obtained by our system, when going from $k = |D|$ to $k = |D|/2$. We report all numbers as percentages.

Dataset	System	All $\|D\|$	Gramm.	Lex ($\|D\|/2$)	Tmpl ($\|D\|/2$)	Random ($\|D\|/2$)
SingleEq	(Koncel-Kedziorski et al., 2015)	72.2	74.2	63.6	67.0	67.2
SimulEq-S	(Kushman et al., 2014)	68.7	70.3	61.1	59.9	66.8

Table 2: Effect of lexical, template overlap and grammaticality data characteristics over various systems.

This again is an instance of remote clique problem, and we follow a similar greedy strategy. We define $f_{\text{tmpl}}(p, D') = \sum_{q \in D'} \textit{PairTempl}(p, q)$, which describes the overlap between a problem p and dataset D'. The greedy method to generate subset D' of D is as follows: we iteratively add a problem p from $D \setminus D'$ to D' which minimizes the value of $f_{\text{tmpl}}(p, D')$. In other words, the algorithm iteratively chooses problem p from $D \setminus D'$ and adds it to D' such that (if possible) p adds a new template to D'. If there are no new templates to be added, then a p whose template is least frequent among those of D' is selected. This process is repeated until D' has k problems.

4.3 Template/Lexical Interaction

Given a dataset and a target subset size k, we would like a subset with the best lexical and template overlap characteristics. As an approximation, we reduce the arithmetic mean of the two overlap values. This mean can be expressed as $H(D) = \frac{1}{2}(\text{Lex}(D) + \text{Tmpl}(D))$. A similar greedy iterative approach as before generates the required dataset with the aforementioned approximation guarantee.

4.4 Grammaticality

Given a dataset, MAWPS can automatically remove ungrammatical word problems, resulting in a subset with improved grammaticality. Our system uses the broad-coverage, linguistically precise English Resource Grammar (ERG) (Flickinger, 2000; Flickinger, 2011) for determining grammaticality. The ERG is a continually-developed implementation of the grammatical theory of Head-Driven Phrase Structure Grammar that covers an increasingly wide variety of phenomena. Moreover, the ERG makes binary decisions for sentence acceptability: if it cannot find a valid parse among its rules, it returns no parse for that sentence. This contrasts with statistical parsers whose utility for this task is limited by the difficulties of tuning a cutoff confidence threshold for acceptability. We allow for filtering problems with ungrammatical sentences.

5 Properties of Current Data

We initialize our repository by extending the publicly available datasets. We list them below in order of increasing complexity of the final equation form.

- **AddSub**: Collection of addition, subtraction problems (Hosseini et al., 2014).
- **SingleOp**: Collection of single operation arithmetic problems (Roy et al., 2015).
- **MultiArith**: Collection of multi-step arithmetic problems (Roy and Roth, 2015).
- **SingleEq**: Single equation problems from (Koncel-Kedziorski et al., 2015).
- **SimulEq-S**: Single and two equation word problems from (Kushman et al., 2014).
- **SimulEq-L**: Collection of single and multiple equation word problems, superset of **SimulEq-S**.

Table 1 shows the characteristics of the datasets under various conditions of grammaticality, lexi-

cal and template overlap. Template overlap varies greatly across datasets, and decreases sharply with increase of complexity of target equation. Lexical overlap, on the other hand, does not show much variation, with most datasets having overlap value of around 7%. Our system effectively reduces overlap, obtaining around 20% reduction in most cases (going from $k = |D|$ to $k = |D|/2$). In the case of template overlap, this reduction varies greatly based on the complexity of target equation, ranging from 2.8% for SingleOp to 97% for SimulEq-L.

As no current system can process **SimulEq-L**, we report the performance of the two most general systems (Koncel-Kedziorski et al., 2015; Kushman et al., 2014) on **SingleEq** and **SimulEq-S** under different overlap and grammaticality properties in Table 2. We use a 5-fold cross validation setup and report average accuracy across folds. We include performance on a randomly selected dataset of the same size to analyze the effect of decreased data-size on performance.

Both systems gain from the pruning of grammatically incorrect problems from the dataset. Our results support the findings of Koncel-Kedziorski et al. (2015) regarding the diminishing performance of systems when faced with reduced lexical and template overlap. We expect, as that work showed, that further reducing these characteristics would result in even lower performance. The results on a random subset of the data show that data set size does not have as strong of an effect on algorithm performance as lexical or template overlap.

6 Conclusion

In this paper, we introduced MAWPS, a repository of math word problems. MAWPS offers a large collection of data from new and published datasets, provides interfaces for adding data, and includes data optimization tools. As current results show, designing general algorithms that can address different problem types while still being robust to the template or lexical variations has remained a challenge. We hope that our framework will help facilitate comparison between results while allowing researchers to focus on targets such as all grammatical problems or all single equation problems. We also include an official 80/20 test/train split of all problems available at the time of this publication.

Acknowledgments: This research was supported by the Allen Institute for AI (66-9175), Allen Distinguished Investigator Award, NSF (IIS-1352249), DARPA (FA8750-13-2-0008) and a Google research faculty award. We thank the anonymous reviewers for their helpful comments.

References

Benjamin Birnbaum and Kenneth J. Goldman. 2007. An Improved Analysis for a Greedy Remote-Clique Algorithm Using Factor-Revealing LPs. *Algorithmica*, 55(1):42–59.

Edward A Feigenbaum, Julian Feldman, and Paul Armer. 1995. *Computers and Thought*. AAAI press.

Dan Flickinger. 2000. On building a more effcient grammar by exploiting types. *Natural Language Engineering*, 6(01):15–28.

Dan Flickinger. 2011. Accuracy vs. Robustness in Grammar Engineering. *Language from a cognitive perspective: Grammar, usage, and processing*, pages 31–50.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to Solve Arithmetic Word Problems with Verb Categorization. In *EMNLP*, pages 523–533.

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Ang. 2015. Parsing Algebraic Word Problems into Equations. *TACL*, 3:585–597.

Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. 2014. Learning to Automatically Solve Algebra Word Problems. In *ACL*, pages 271–281.

Arindam Mitra and Chitta Baral. 2015. Learning to automatically solve logic grid puzzles. In *EMNLP*.

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text. In *EMNLP*, volume 1, page 2.

Subhro Roy and Dan Roth. 2015. Solving General Arithmetic Word Problems. In *EMNLP*.

Subhro Roy, Tim Vieira, and Dan Roth. 2015. Reasoning about Quantities in Natural Language. *TACL*.

Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. 2015. Solving Geometry Problems: Combining Text and Diagram Interpretation. In *EMNLP*.

Shuming Shi, Yuehui Wang, Chin-Yew Lin, Xiaojiang Liu, and Yong Rui. 2015. Automatically Solving Number Word Problems by Semantic Parsing and Reasoning. In *EMNLP*.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015. Towards AI-complete Question Answering: a set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Lipu Zhou, Shuaixiang Dai, and Liwei Chen. 2015. Learn to Solve Algebra Word Problems Using Quadratic Programming. In *EMNLP*.