Mathematical Word Problem Solution Evaluation via Data Preprocessing Approach

Andrii D. Nikolaiev and Anatoliy V. Anisimov

Taras Shevchenko National University of Kyiv, 4d, Glushkova str., Kyiv, 8300, Ukraine

Abstract

This article describes an overview of methods for processing mathematical text problems and correspondent domain datasets. It was proposed a new approach to estimate MWP solutions that could use more context around the problem. Described how method could be used for MWP similarity and for automation mathematical solutions grading.

Keywords [1]

Math word problem, natural language processing, machine learning, artificial intelligence

1. Introduction

The importance of personalized learning has always been in high demand. But for quality learning it is necessary to have an expert – a person or a system for self-control. There are several ways to estimate level of knowledge, the most common is with the help of multiple-choice tests. Modern mathematical platforms are limited in the ability to accept tasks and mostly capable only to accept answers for MCQ-type questions (Multiple Choice Question), which are about to choose between only correct and incorrect answers from the proposed ones. Despite the fact, this approach provides an opportunity to objectively compare student results, it is not objective for estimation of comprehension of subject.

More indicative is the concept inventories. Unfortunately, there are no significant automatic mathematical word problem solvers and estimators. It imposes high limitation for the possibilities in learning new mathematical concepts and detailed feedback loss over the learning process. Therefore, it is important to be able to accept detailed solutions over mathematical problems automatically.

The problem is particularly challenging because a wide semantic gap remains between the human-readable words and machine understandable logics. It is also hard to provide relevant feedback about the given solution. That's why we need to dive into some well-known NLP problems connected with the mathematical text interpretation. The one of them is designing an automatic solver for the mathematical word problem (MWP) which started back in 1960's, and in the last years due to scientifically increased interest to artificial intelligence (AI) became more popular.

Formalization systems could also be applied for solving and understanding mathematical problems (e.g., proof assistant LEAN [1], SAD [2]). The proof assistant is a piece of software that provides a language for defining objects, specifying properties of these objects, and proving that these specifications hold. The system checks that these proofs are correct down to their logical foundation. These tools are often used to verify the correctness of programs. But they can be also used for abstract mathematics. In the formalization, all definitions are precisely specified and all proofs are virtually guaranteed to be correct. The International Mathematical Olympiad Grand Challenge was announced in 2020 [3, 4]. The challenge is about creating an AI that can win a gold medal at the IMO. To remove ambiguity of the scoring rules, teams were proposed

Information Technology and Implementation (IT&I-2021), December 01-03, 2021, Kyiv, Ukraine
EMAIL: nikolaev@knu.ua (A. 1), a.v.anisimov@knu.ua (A. 2)
ORCID: 0000-0002-1467-2006 (A. 2)

the formal-to-formal (F2F) variant of the IMO: the AI receives a formal representation of the problem (in the LEAN Theorem Prover), and it is required to emit a formal (i.e., machine-checkable) proof. However, this way is less practical because of complexity of describing problem statement and check the correctness of the given solution via such systems.

The remainder of the paper is organized as follows. Paper starts with the review of the AWP solvers in Section 2, followed by problem statement and method implementation main steps in Section 3. Since dataset deserves special attention, it was summarized as well as the associated ideas of dataset extension in Section 4. The general algorithm of the proposed method is described in Section 5. In the section was also described related ideas about similarity to math problems and variants of method usage for real case situations. The paper concludes list of references in the final sections.

2. Related work

MWP is the common math problem type used for building solvers. There are some categories of MWP: Arithmetic Word Problem (AWP), Equation Set (ESS), Geometric Word Problem (GWP).

We will take a closer look at AWP type problems which are due to given textual condition of the problem is about to compose a corresponding equation and get the answer to the problem. The problem is represented by a sequence of words $\langle w_0, w_1, \dots w_k \rangle$ – some of them are quantities $q_0, q_1, \dots q_n$, which are mentioned in the text and the unknown variable x. The main goal is to present a text problem in the form of the corresponding equation E which is linear for the case. Among the basic arithmetic operations there are only $\{+, -, \times, \div\}$.

Table 1
An example of arithmetic word problem

MATH WORD PROBLEM
Misha found 221 seashells and 35 starfish on the beach. He gave 101 of the seashells to Katya. How many seashells does Misha now have?
OUTPUT EQUATION: x = 221 + 35 – 101 SOLVER OUTPUT: 155

The different approaches which have been invested in solving math word problems can be categorized into four main efforts: (1) statistics-based, (2) tree-based, (3) deep learning-based and rule-based methods which were used on the early approaches which we will skip for now.

Some approaches used a combination of the above categories.

2.1 Statistic-based Approach

The statistic-based approach tries to identify the question entities, their values and the desired operators that need to be evaluated in order to achieve the correct answer. The identifications are obtained with common machine learning methods. Hosseini et al. 2014 suggested solving arithmetic word problems which include addition and subtraction operations [5]. The problem text is split into parts where each represents a transition between two world states. The quantities of the entities for each such transition are updated or observed, and the predicted solution is inferred from changing the world states until reaching the end of transitions. Solving Math Word Problems Using Encoder-Decoder Neural Network Mitra &

Baral (2016) used supervised learning for specifying the formula that should be applied to generate the appropriate equation and the relevant variables [6]. Liang et al. showed a similar work using log-linear models with handcrafted feature engineering [7].

2.2 The Tree-based Approach

The tree-based approach focuses on representing the problem in hierarchical manner. The hierarchy is represented by a unique tree named binary expression tree. An expression tree can be evaluated by applying the operator at the root to the values obtained by recursively evaluating the left and right subtrees. Koncel-Kedziorski et al. (2015) suggested a system named ALGES which generates a tree over the space of all valid expression trees, given a math word problem with single equation formula as the answer [8]. It uses integer linear programming and maximum-likelihood estimator. Roy & Roth (2016) presented the Expression Tree method [9]. It uniquely decomposes the math problem into multiple classification problems and then composes a monotone expression tree, which defines a collection of simple prediction problems, each determining the lowest common ancestor operation between a pair of quantities mentioned in the problem. Roy & Roth (2017) defined a new structure named Unit Dependency Graph (UNITDEP), an annotated graph with vertices for each of the quantities appearing in the problem and edges representing the relationship between two quantities. The graph is annotated by classifiers for node labeling and edge properties annotating [10].

2.3 Deep Learning-based Approach

In the deep learning approach, MathDQN by Wang et al., (2018) is a customized form of the general deep reinforcement learning framework. They defined actions, states and a reward function, and used a feed-forward network [11]. TheSeq2Seq model proposed by Wang et al. (2017) includes an encoder (a GRU unit) and a decoder (an LSTM unit) producing an equation template [12]. T-RNN (Wang et al., 2019) also extends the Seq2Seq model. It encodes the quantities using Bi-LSTM and self-attention network. It uses an RNN to construct a tree-structure template with inferred numbers as the leaf nodes and unknown operators as the inner nodes. The tree-structure template is designed to reduce the number of template space [13]. Amini et al. (2019) suggested the Sequence2Program model for multiple-choice math word problems. It uses an encoder-decoder neural network that maps word problems to a set of feasible operation programs. The results of the executed operation program are matched against the given multiple-choice options for a particular problem [14].

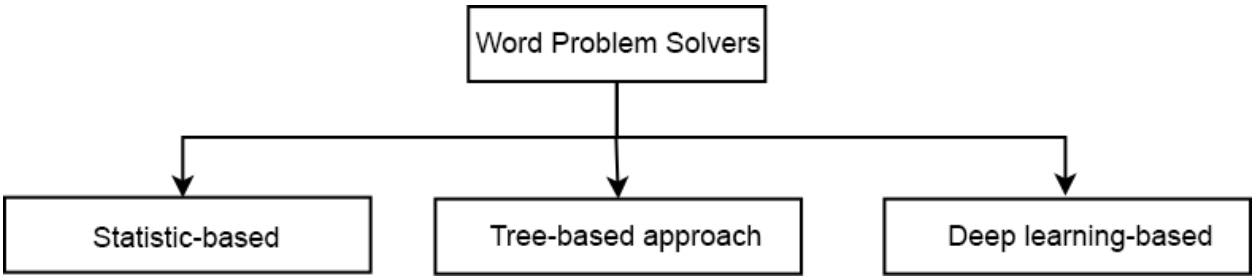

Figure 1: Main word problem solvers categories.

3. Problem statement

The aim of the work is to propose an algorithm that is able to maintain significant points on the given solution for the mathematical text problem and provide the verdict.

For input algorithm takes solution sample of the given problem statement. Algorithm output is to return solution score (0 or 1) and maintain solution part which is led to the decision (provide keywords or crucial points for qualitive estimation).

3.1 Method implementation main steps

In order to implement method, it is important to complete all the steps below:

1. **Algorithm**. Machine learning methods could be useful for the problem. The approach uses n-gram extraction as the main source of comparison of solutions that could be inaccurate. At the same time, using phrase embeddings to resolve such cases as paraphrases is promising.
2. **Dataset**. One of the most important steps in method implementation is to create dataset of type *Problem-Solution-Result*. It could help to understand the way of solutions evaluation for future work.
3. **Evaluation metrics**. To develop evaluation metrics for a given task. This should be a vital part as the task is non-standard and it's not obvious how to compare different systems. The same equation could be formulated in different ways. While for the standard MWPs F1 score is OK there is no such metrics for estimation of detailed solutions.
4. **Methods leaderboard**. Leaderboard is highly convenient and presentable for better comparison over other methods in future studies. Adding scores/results of described methods and their implementation versions should help in evaluation and estimation of provided different methods.
5. **Module implementation.** Module needs to be connected to database so it could upload a new solution and estimate it in correspondence to known verdicts. The module also could bring some useful insights on how the "ideal" solution could look like. Therefore, module itself could be used as generator for problem solutions and be used for finding similarities between the problems statement.
6. **Experiments and discussion part.** After module implementation, we need to test proposed method solution, evaluate it and compare with well-known approaches using NLP methods. Module could also be tested over the real case situations as it could be used in education field for correspondent math classes in automation processes.

In this paper was provided first version of algorithm via machine learning methods, overview of dataset and some basic ideas of how the proposed algorithm could use for MWP similarity and for real case situations.

4. Datasets

For providing solution for a given type of problem we could obtain semantic part via some NLP-models. There are bunch of datasets which is used for the providing solutions to MWP.

For the experiments before 2016 commonly used datasets were [15, 16]:

- **Alg514** (Kushman et al., 2014) The dataset is crawled from algebra.com, a crowd-sourced tutoring website and contains only 514 linear algebra problems with 28 equation templates.
- **AI2** (Hosseini et al., 2014): There are 395 single-step or multi-step arithmetic word problems for the third, fourth, and fifth graders. It involves problems that can be solved with only addition and subtraction. The dataset is harvested from two websites: math-aids.com and ixl.com.
- **Dolphin1878** (Shi et al., 2015) includes 1,878 number word problems with 1183 equation templates, obtained from algebra.com and Yahoo! answers.
- **DRAW** (Upadhyay and Chang, 2016) containing 1,000 algebra word problems from algebra.com, each annotated with linear equations.
- **SingleEQ** (by Koncel-Kedziorski et al., 2015) The dataset contains both single-step and multi-step arithmetic problems and is a mixture of problems from a number of sources, including math-aids.com, k5learning.com, ixl.com and a subset of the data from AI2. Each problem involves operators of multiplication, division, subtraction, and addition over non-negative rational numbers.
- **Dolphin18K** which contains over 18,000 annotated math word problems. It is constructed by semi-automatically extracting problems, equation systems and answers from community question-answering (CQA) web pages. The source data we leverage are the (question, answer, text) pairs in the math category of Yahoo! answers.

- **MAWPS** is another testbed for arithmetic word problems with one unknown variable in the question. Its objective is to compile a dataset of varying complexity from different websites. Operationally, it combines the published word problem datasets used in AI2 and some others. There are 2,373 questions in the harvested dataset [17].
- In 2017 Wang et al. presented the **Math23K** dataset. The dataset contains Chinese math word problems for elementary school students and is crawled from multiple online education websites [12].

The latest datasets are:

- **Ape210K** (2020): 210K Chinese elementary school-level math problems [18].
- In 2021 Arkil at el. used models Graph2Tree with RoBERTa, GTS with RoBERTa, LSTM Seq2Seq with RoBERTa, Transformer with RoBERTa on the data of a presented **SVAMP** dataset, which contained 1000 problems with compiling elementary school level equations created by applying carefully chosen variations over examples sampled from existing datasets [19].
- In 2021 Dan et al. used a GPT-3 model based on **MATH** dataset which consists of 12,500 from high school math competitions which was proposed in the work and also contains solution base for each problem [20] and **ASMP** pretraining corpus, which consists of Khan Academy and Mathematica data. AMPS has over 100,000 Khan Academy problems with step-by-step solutions in LaTeX. It also contains over 5 million problems generated using Mathematica scripts, based on 100 hand-designed modules covering topics such as conic sections, div grad and curl, KL divergence, eigenvalues, polyhedra, and Diophantine equations. In total AMPS contains 23GB of problems and solutions.

Yet, none of datasets contains one or more alternative solutions for every problem sample. Therefore, in order to run experiments for future method it's essential to create *Problem-Solution-Result* dataset. Hopefully, we already have first such corpuses (as ASMP) and we could expect even more big corpuses as, for instance, in 2015 was mentioned that Baidu has collected over 950 million questions and solutions in its database for K-21 [16]. Additionally, for our dataset we could use the information of some well-known sites and databases where problems are stored with solutions refer to them, e.g.:

- Use some of fully solved shortlists of recent IMO competitions which are available on the official IMO web-site, while the solutions of all shortlisted problems from 1959-2009 are available in The IMO Compendium [21].
- Export and use anonymous data from messenger chat history which are used to process problem solutions of refer mathematical classes as far as more schools switched to online format.

5. Solution Preprocessing Approach

5.1 Solutions Context for Semantic Extraction

It is difficult to understand the semantics behind the problem statement only. So, in order to simplify the task, we can use more context around the problem.

For the context we could use other solutions (both incorrect and correct ones) and their estimation scores for them.

5.2 The Algorithm for Solution Evaluation

For the given text problem W_0, we have M solutions $S = \langle S_0, S_1, \dots S_M \rangle$, each solution could be presented with their own sequence of words $w = \langle w_0, w_1, \dots w_k \rangle$ – some of them are quantities $q_0, q_1, \dots q_n$. Every solution corresponds some binary vector of results r_i: 0 refer to incorrect solution and 1 – to the correct one. We use one of mentioned models to extract vector of keywords (or phrases – together main features for a given problem W_0) $k = \langle k_0, k_1, \dots k_l \rangle$ from all solutions data. We train our model on correlation between k and r. Every keyword could be *positive* and *negative* depending on the correlation with r. Then for each value k_i we obtain the F1-score k_i^{F1} of how often the specific keyword resulted in correct

evaluation of a given solution plus sign-factor k_i^{sign} which is 1 in case of positive keyword and -1 – for negative keyword. The final vector k sorted by key = $F1(k_i)$ and W form set of parameters for our evaluator $eval_w$: $\langle W, k \rangle$.

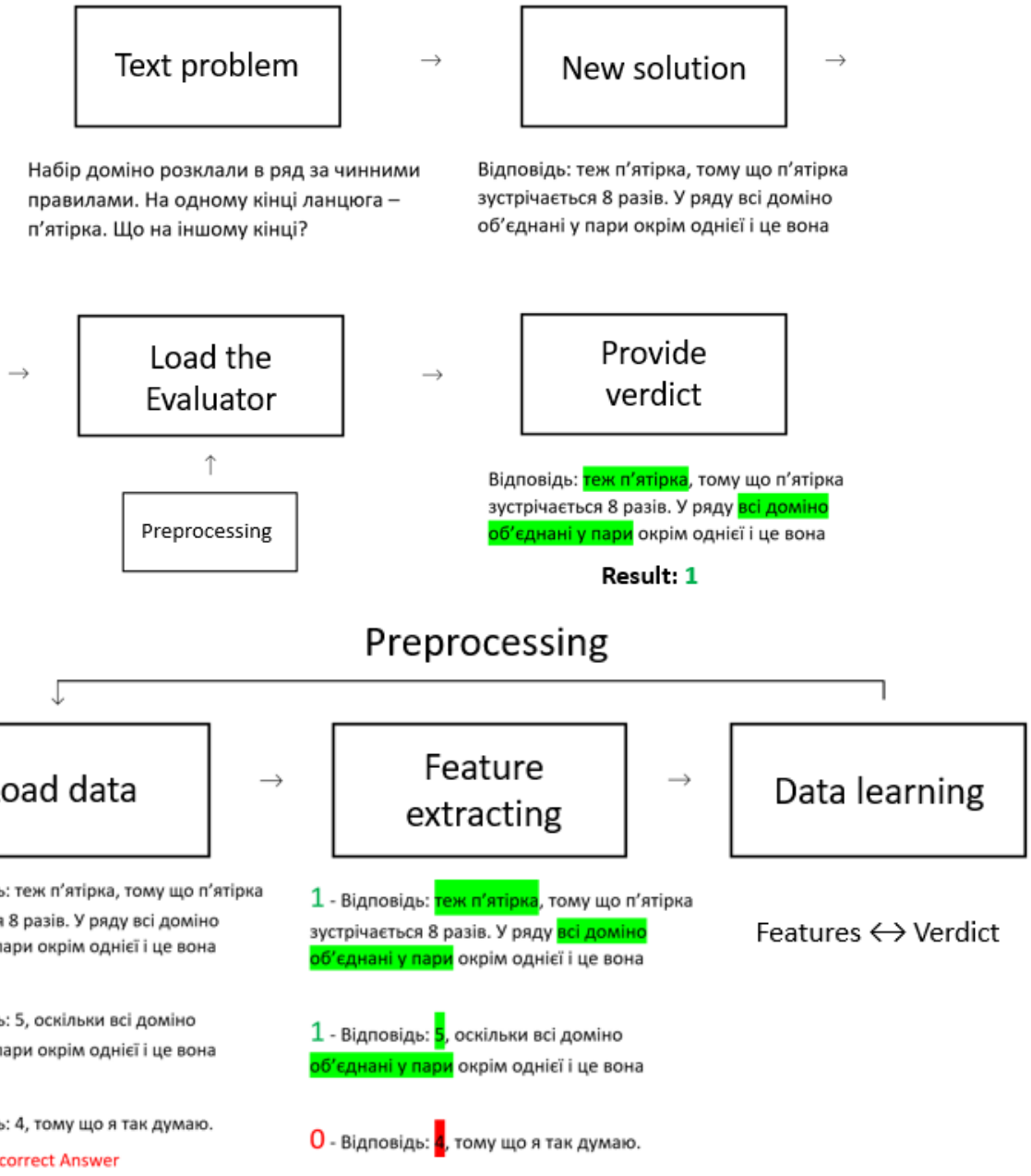

Figure 2: The approach using solutions context for providing the evaluation over the given solution (in Ukrainian)

Now we obtain some example solution $S' = \langle w_0, w_1, \ldots w_k \rangle$ in which we find presence of all the keywords from k and obtain the vector p, where p_i is the binary value of presence keyword k_i in the given solution: 0 in case S' not include k_i, 1 – either.

Finally, we calculate the weighted sum of our vector p where the weights are corresponding F1-scores and give a verdict r_S, for the solution via some activation function (for instance, binary step activation function $H(x)$ as far as we could provide verdict only from $\{0, 1\}$ set).

Note that it is important to give more penalties for the solutions which contains negative keywords with high F1-scores as far as correct solutions may not include any mistakes.

On the given problem set we could now generate $eval_w = \langle W, k \rangle$ for each problem.

5.3 MWP Similarity

Table 2
Math word problems similarity

MATH WORD PROBLEM A		SIMILAR MATH WORD PROBLEM B
Masha had 5 fruits: 3 apples and 2 bananas. How many different ways to eat one fruit per each day during 5 days?		We have 5 letters: three letters "A" and two letters "B". How many different 5-letter words could be created by using all given letters?
OUTPUT SOLUTION: Let's encode the choices: "A" for apples and "B" for bananas, "AAABB" will stand for the case when Masha takes 3 apples at first 3 days and then 2 bananas for the rest 2 days. Therefore, we need to calculate all permutations of the word "AAABB" <…>. The answer is $\frac{5!}{3!2!}$ SOLVER OUTPUT: $\frac{5!}{3!2!}$	↔	OUTPUT SOLUTION: Imagine if we have 5 different letters and the same question. Then the answer would be 5!. But we need not to count the repetitions. Therefore, the answer needs to be divided by number of permutations among three letters "A" and two letters "B", which is 3! and 2! respectively. So, the answer is $\frac{5!}{3!2!}$ SOLVER OUTPUT: $\frac{5!}{3!2!}$

It could be hard to understand problem similarity via the statement. Instead, proposed to find the relation between $eval_a$ and $eval_b$ of the problems. Hence, to understand the similarity between two given problems and solutions we need to compare their k vectors plus their parameters and W.

Solution similarity helps with semantic problem understanding and also help in classifying the problems into smaller classes.

Table 3
Table of corresponded phrases

Problem A	Problem B
5 fruits	5 letters
3 apples	three letters "A"
2 bananas	two letters "B"
5 days	5-letter words

5.4 Solutions Preprocessing for MWP

As far the MWP mostly about composing appropriate equations based on the text in order to provide the correct answer, the solutions preprocessing approach can't be used directly.

However, the solutions context could be applied for MWP. For instance, in the given example there was

an error: the correct answer is 120 and correct equation is “x = 221 – 101”. That is because model has the statement incorrect interpretation.

Table 4
An example of correctly solved arithmetic word problem

MATH WORD PROBLEM		SIMILAR MATH WORD PROBLEM
Misha found 221 seashells and 35 starfish on the beach. He gave 101 of the seashells to Katia. How many seashells does Misha now have?	↔	Vasya count 10 red cars and 3 green cars while was walking to home. Then he saw 5 more red cars from the window. How many red cars did he saw?
OUTPUT EQUATION: x = 221 + 35 – 101 SOLVER OUTPUT: 155 CORRECT EQUATION: x = 221 – 101 CORRECT OUTPUT: 120		OUTPUT EQUATION: x = 10 + 5 SOLVER OUTPUT: x = 15

Such error could be tracked via use of solutions preprocessing method. Over some data of correctly (and incorrectly) solved MWP, we can obtain features which could be generalized for the similar task problems. The “similarity” could be reached over large amount of *Problem-and-Solutions* data with evaluators $eval_w$ similarity (5.3). Over the big amount of data, for the given example it would be not difficult to see that “35 starfish” is irrelevant data because the similar solutions do not include it in solutions as far as the problem question does not contain word “starfish”.

5.5. Method usage perspectives for real case situations

Such system could be effectively applied for automated checking of school children mathematical solutions. In order to use the system in real case situations it is also important to use some recognition techniques and learn our method to differ where is the problem statement and solution is placed. After we get the text, we could process it through the mentioned method and obtain evaluation results.

Of course, for the given task it’s important to obtain high quality data what would be difficult in refer to pupils. However, the practical usage of the system without no doubt is highly promising.

6. Conclusion

It was done an overview of well-known methods for MWP and correspondent domain datasets. Also, it was proposed a new approach to estimate MWP solutions and described an idea of how module could be used for MWP similarity. The proposed method could be used for automation mathematical solutions grading.

The last works approaches about MWP reached a high degree of accuracy [22, 23, 24], however, for some other datasets (such as SVAMP) the majority of models can’t provide significant results. That is because of semantic gap as natural language texts invariably assume some knowledge implicitly or information noise. Humans know the relevant information, but a computer reasoning from texts must be

given it explicitly. Filling these information gaps is a serious challenge; representation and acquisition of the necessary background knowledge are very hard AI problems.

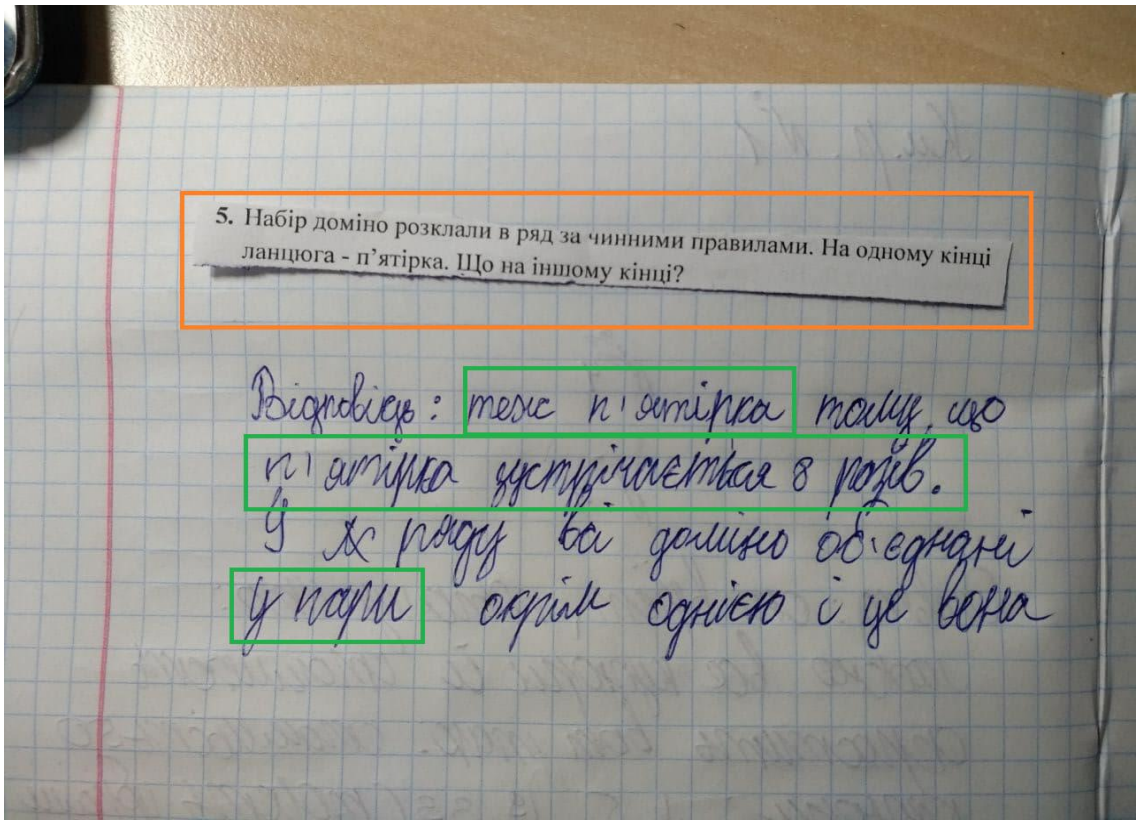

Figure 3: The model extracting main features from the student solution (in Ukrainian). The orange allocation is where the solution statement is placed, while green ones are the main points of the correct solution. System will grade given solution as the correct one

Therefore, there is a big necessity in having more detailed datasets of not only problems with correct answers, but with suggested solutions part too. With the help of given algorithm using preprocessing approach to the problem sets, our method could use more context around the problem and increase accuracy in MWP and even more difficult (such as mathematical olympiad) problems.

7. References

[1] Lean system proofer. URL https://leanprover.github.io/about/

[2] Verchinine, Konstantin & Lyaletski, Alexander & Paskevich, Andrey & Anisimov, A.. (2008). On Correctness of Mathematical Texts from a Logical and Practical Point of View.. 583-598.

[3] IMO Grand Challenge. URL https://imo-grand-challenge.github.io/

[4] The International Mathematical Olympiad (IMO). URL https://www.imo-official.org/

[5] Hosseini, M. J., Hajishirzi, H., Etzioni, O., and Kushman,N. Learning to solve arithmetic word problems with verb categorization. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 523–533, 2014.

[6] Mitra, A. and Baral, C. Learning to use formulas to solve simple arithmetic problems. In Proceedings of the 54thAnnual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2144–2153,2016.

[7] Liang, C.-C., Wong, Y.-S., Lin, Y.-C., and Su, K.-Y. A meaning-based statistical english math word

problem solver. arXiv preprint arXiv:1803.06064, 2018.
[8] Koncel-Kedziorski, R., Hajishirzi, H., Sabharwal, A., Etzioni, O., and Ang, S. D. Parsing algebraic word problems into equations. Transactions of the Association for Computational Linguistics, 3:585–597, 2015.
[9] Roy, S. and Roth, D. Solving general arithmetic word problems. arXiv preprint arXiv:1608.01413, 2016.
[10] Roy, S. and Roth, D. Unit dependency graph and its application to arithmetic word problem solving. In Thirty-First AAAI Conference on Artificial Intelligence, 2017.
[11] Wang, L., Zhang, D., Gao, L., Song, J., Guo, L., and Shen,H. T. Mathdqn: Solving arithmetic word problems via deep reinforcement learning. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[12] Wang, Y., Liu, X., and Shi, S. Deep neural solver for math word problems. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 845–854, 2017.
[13] Wang, L., Zhang, D., Zhang, J., Xu, X., Gao, L., Dai, B., and Shen, H. T. Template-based math word problem solvers with recursive neural networks. 2019.
[14] Amini, A., Gabriel, S., Lin, P., Koncel-Kedziorski, R., Choi,Y., and Hajishirzi, H. Math qa: Towards interpretable math word problem solving with operation-based formalisms.
[15] Danqing Huang Shuming Shi Chin-Yew Lin Jian Yin Wei-Ying Ma. "How well do Computers Solve Math Word Problems? Large-Scale Dataset Construction and Evaluation". Meeting of the Association for Computational Linguistics (2016).
[16] Dongxiang Zhang, Lei Wang, Luming Zhang, Bing Tian Dai and Heng Tao Shen. "The Gap of Semantic Parsing: A Survey on Automatic Math Word Problem Solver". URL shttps://arxiv.org/pdf/1808.07290.pdf
[17] Rik Koncel-Kedziorski, Subhro Roy, Aida Amini1, Nate Kushman, Hannaneh Hajishirzi1. "MAWPS: A Math Word Problem Repository."
[18] Wei Zhao, Mingyue Shang, Yang Liu, Liang Wang, Jingming Liu. "Ape210K: A Large-Scale and Template-Rich Dataset of Math Word Problems."
[19] Arkil Patel, Satwik Bhattamishra, Navin Goya. "Are NLP Models really able to Solve Simple Math Word Problems?" Available at: https://arxiv.org/pdf/2103.07191v2.pdf
[20] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, Jacob Steinhardt. "Measuring Mathematical Problem Solving With the MATH Dataset". URL: https://arxiv.org/pdf/2103.03874v1.pdf
[21] Filip Maric, Sana Stojanovic. "Formalizing IMO Problems and Solutions in Isabelle/HOL" https://arxiv.org/abs/2010.16015
[22] Jipeng Zhang, Lei Wang, Roy Ka-Wei Lee, Yi Bin, Yan Wang, Jie Shao, Ee-Peng Lim, "Graph-to-Tree Learning for Solving Math Word Problems". URL: https://www.aclweb.org/anthology/2020.acl-main.362.pdf
[23] Zhipeng Xie, Shichao Sun. A Goal-Driven Tree-Structured Neural Model for Math Word Problems. URL: https://www.ijcai.org/Proceedings/2019/0736.pdf
[24] Jierui Li, Lei Wang, Jipeng Zhang, Yan Wang, Bing Tian Dai, Dongxiang Zhang. Modeling Intra-Relation in Math Word Problems with Different Functional Multi-Head Attentions. URL: https://www.aclweb.org/anthology/P19-1619.pdf