IJARSCT

ISSN (Online) 2581-9429

Impact Factor: 7.301

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Volume 3, Issue 1, May 2023

Math Word Problem Solver using Machine Learning

Bhuvanesh Patil[1], Gauri Mahore[2], Ankit Kumar[3], Saurabh Varule[4], Prof. Laxman Pawar[5]

Student, Department of Computer Engineering[1,2,3,4]

Professor, Department of Computer Engineering[5]

Sinhgad College of Engineering, Pune, Maharashtra, India

Abstract: *This paper presents a deep neural solver to automatically solve math word problems. In contrast to previous statistical learning approaches, we directly translate math word problems to equation templates using a recurrent neural network (RNN) model, without sophisticated feature engineering. We further design a hybrid model that combines the RNN model and a similarity-based retrieval model to achieve additional performance improvement. Experiments conducted on a large dataset show that the RNN model and the hybrid model significantly outperform state of-the-art statistical learning methods for math word problem solving.*

Keywords: RNN, LSTM, MWPs, DOLPHIN 18K, GSM8K

I. 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 test bed for algorithms that build a precise understanding of what is being asserted in text. MWPs solving is believed to be challenging because of the semantic gap between the mathematical expressions and language logic. We explored different deep learning methods with least amount of feature engineering.

Our works involves the four main steps. First, we preprocess our datasets. Second, three different models are built, including a bidirectional LSTM (encoder)-LSTM (decoder) attention model, a bidirectional GRU (encoder)-LSTM (decoder) attention model, and a transformer model. Third, we tune some hyperparameters. Last, we carried out qualitative analysis of our generated outputs to study the behavior of our model.

II. PROBLEM STATEMENT

MWPs solving is believed to be challenging because of the semantic gap between the mathematical expressions and language logic [5]. The ordering in the mathematical expressions don't always matter as what is expected in simple text NLP problem because of commutative laws of addition and multiplication. Because of this, the state-of-the-art of MWPs only achieve 36% accuracy in a 5-choose-1 multiple choice task [5]. In this paper, we focused our work on arithmetic mathematical problems, and constructed models to output the mathematical equations given a chunk of MWPs in English text.

III. RELATED WORK

The inspiration for creating MWPs came from observing parents of elementary school kids spending significant time every day to check their kids' homework and tests. We thought that MWPs will be of immense help to the parents, if it can scan the kids' word problem and output an equation and final solution. Further, as mentioned by (Zhang et al., 2019), designing an automatic solver for mathematical word problems has a long history dating back to the 1960s (Bobrow, 1964), (Feigenbaum et al., 1963), (Charniak, 1969). The problem is particularly challenging because there remains a wide semantic gap to parse the human-readable words into machine-understandable logic so as to facilitate quantitative reasoning. Hence, math word problem solvers are broadly considered as good test beds to evaluate the intelligence level of agents in terms of natural language understanding (Clark, 2015), (Clark & Etzioni, 2016) and the successful solving of math word problem solvers would constitute a milestone towards general artificial intelligence. The work done by (Wang et al., 2017), and recently by (Sizhu Cheng, 2019) is of particular interest to our approach to using deep neural networks in solving math word problems. We delved deeper on work done by (Sizhu Cheng, 2019), using it as guidance for baseline for our project.

IV. PROPOSED MODEL

We began by doing an extensive survey of the existing literature and related work done in, which gave an extensive overview of several work which has been done in the field of math word problem solvers.

The proposed system has three main operational aspects each having different feasibility associated with them. The process of performing an operation after the operator has been predicted is a P problem. The main algorithm to develop the word problem solver system is Recurrent Neural Network (RNN) which is an advanced variation of artificial neural network used for processing sequential data. The network needs to be trained before it can solve the input problem. The process of training the network can be concluded as NP-complete problem from.

The main result of this paper is that the training problem for the Neural Network is NP. complete. That is, unless P= NP there is no polynomial-time algorithm that given a collection of training examples on n Boolean inputs, can always correctly decide whether there exist linear threshold functions for nodes N1, N2, and N3 so that the Neural Network produces output consistent with the training examples. Training can be done in polynomial- time using linear programming techniques. That is, unless P NP there is no polynomial- time algorithm that given a collection of training examples on n Boolean inputs, can always correctly decide whether there exist linear threshold functions. However, when the system is given a set of previously unseen word problems it turns out to be an optimization problem.

4.1 Dataset

As a next step, our work involved procuring math word problem datasets. Math Word Problem Repository is a repository with extendable mathematical word problems [3]. It allows people to keep adding new single problem and equips with backend tool to select datasets with reduced lexical overlap. Each piece of data from MAWPS is a dictionary. It may have keys including index, alignments, equations, solutions and questions. However, most of data don’t have all these attributes. We preprocessd the data obtained from MAWPS and collected all data with both questions and solutions to be used for our models. There are a few datasets available for the algebraic problem solving that have been used for both, rule based solvers as well as machine learning based solvers that were used in published literature.(Dolphin 18k,MAWPS,GSM8K) .Eventually carrying out necessary pre-processing on cleaned up datasets. For ex. The Dolphin18k dataset was built using the scripts and tools provided by the authors. The scripts collect the data from Yahoo Answers. Data clean up is to be performed as per the authors's instructions. The cleaned up dataset will be available in a json format. MAWPS is readily available as json file and no webscraping is necessary to build it. Further more, each entry in source and target is converted to lower case to avoid emphasis on uppercase letters in the dataset.

4.2 Preprocessing

This Dolphin18k dataset was built using the scripts and tools provided by the authors. The scripts collect the data from Yahoo Answers. Data clean up was performed as per the authors’s instructions. The train and dev test splits are also provided by the scripts. The cleaned up dataset is available in a json format. Alg514, Draw-1k and MAWPS are readily available as json files and no webscraping is necessary to build them.

Our models, both Bi-LSTIM-Attn and Transformer both require input source file and target file. So, for each example entry, ”text” section from json files is written to .txt and ”equation” section is written to .txt. Further more, each entry in source and target is converted to lower case to avoid emphasis on uppercase letters in the dataset.

A subset of examples were extracted from all the datasets, that contain a single unknown. This was achieved by writing a script that parses the equations entry of the json file, search for number of unknowns and equations, extract only the examples that contain a single unknown.

4.3 Train dev Split

The final goal of our Deep Learning model is to understand and solve elementary math word problems. In order to achieve this, the train/dev set distributions have to be clearly distinguished. The train set can comprise of elementary as well as complex problems but the dev set can only contain elementary problems. In this approach, we split the preprocessed 45446 training data from all three datasets into 9:1 ratio each time and use 10-fold cross validation in our

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 1, May 2023

training. We use tensor2tensor (T2T) library, released and maintained by Google AI team, to perform all the experiments related with transformer [9]. The loss used in this model is the cross entropy loss.

We mix all training data from the three datasets and randomly shuffle them. We created a test set by combining the test set from GSM8K and Dolphin 18k and randomly shuffling them as well (note that MAWPS is a single repository without train-val-test splitting). By this, we obtained 45446 piece of training data and 325 test data in total

4.4 Algorithm

Baseline RNN model model uses a Bidirectional LSTM Encoder and a Unidirectional LSTM Decoder. This Seq2Seq Model has multiplicative attention, as shown on the third step of the decoder. The baseline RNN produces a probability distribution P, over target words at the tth timestep. Here, V; is the size of the target vocabulary. Loss function is shown below as well. Here, A represents all the parameters of the model and Jt(0) is the loss on step t of the decoder. 9t is the 1-hot vector of the target word at timestep t.

$$Pt = Softmax(Wvocab0\ t)$$

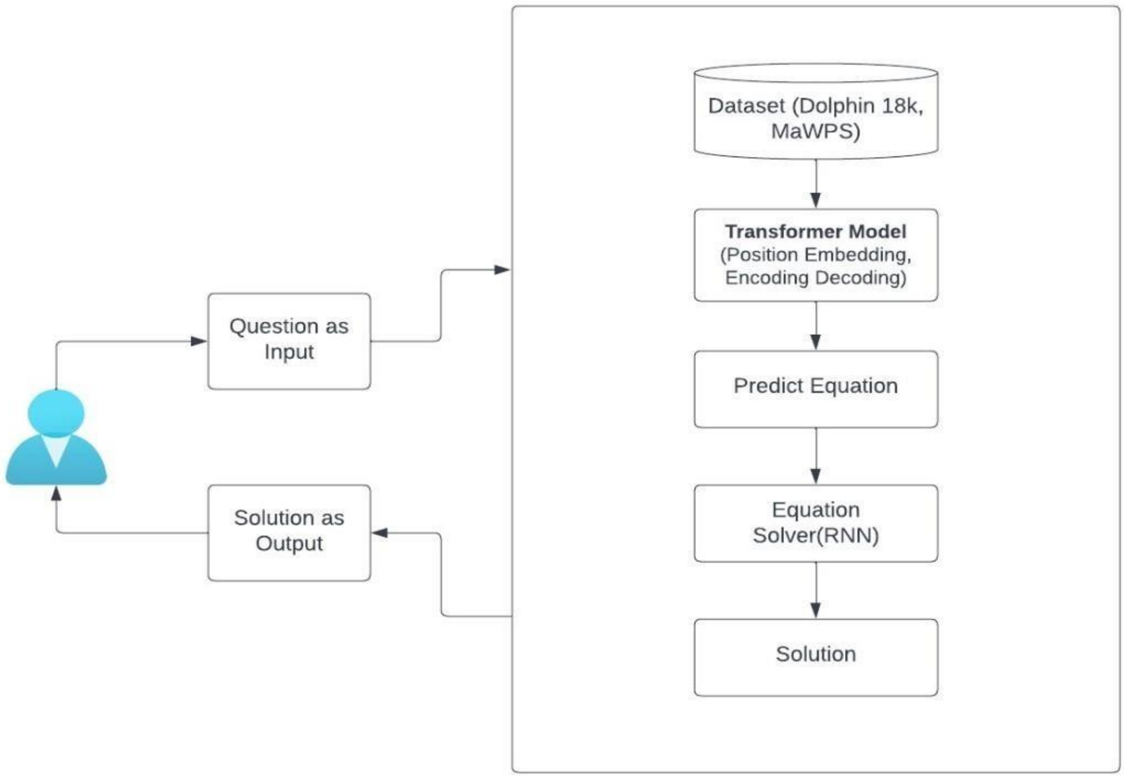
$$where\ Pt \in v1 \times 1, \quad Wvocab \in v1 \times h$$

$$J(\theta) = C\ E\ (Pt, gt)$$

4.5 Transformer

The idea of using transformer models come from identifying MWPs Solving as a seq2seq text input problem. Some recent applications in mathematical reasoning also indicate the validity of using transformer to handle mathematics . We recognize that building a problem solver for a word math problem shares similarities with machine translation in multiple ways. Both tasks involve seq2seq models, and formulating the math problems in equations is analogous to translation from English language into mathematical language. In light of these, we plan to incorporate transformer layers in our experiments and test them to see how they perform in MWPs Solving.

V. SYSTEM ARCHITECTURE DIAGRAM



VI. CONCLUSION

We have proposed an RNN-based seq2seq model to automatically solve math word problems. This model directly transforms problem text to a math equation template. This is the first work of applying deep learning technologies to math word problem solving. In addition, we have designed a hybrid model which combines the seq2seq model and a retrieval model to further improve performance. A large dataset has been constructed for model training and empirical evaluation. Experimental results show that both the seq2seq model and the hybrid model significantly outperform state- of-the-art statistical learning methods in math word problem solving.

The output of our seq2seq model is a single equation containing one unknown variable. Therefore our approach is only applicable to the problems whose solution involves one linear equation of one unknown variable. As future work, we plan to extend our model to be able to generate equation systems and nonlinear equations.

REFERENCES

[1]. Simple Mathematical Word Problems Solving with Deep Learning : Sizhu Cheng, Nicolas Chung-Stanford University Dec 2019

[2]. Math Bot–A Deep Learning based Elementary School Math Word Problem Solver : Anish Kumar Nayak, Rajeev Patwari, Viswanathan Subramanian - Stanford University 2019

[3]. Training Verifiers to Solve Math Word Problems : Karl Cobbe, Vineet Kosaraju, Mohammed Bavarian, Mark Chen, Heewo Jun, Lukass Kaiser, Matthias Pieppert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, John Schluman-Open AI, 18 Nov 2021

[4]. Formal Mathematics Statement Curriculum Learning: Stanislas Polu, Jesse Michael Han, Kunhao Zheng, Mantas Baksys, Igor Babuschkin, Ilya Sutskever - 3 Feb 2022

[5]. A Neural Network Solves, Explains, and Generates University Math Problems by Program Synthesis and Few-Shot Learning at Human Level: Iddo Drori, Sarah Zhanga , Reece Shuttlewortha, Leonard Tangc , Albert Lua , Elizabeth Kea , Kevin Liua , Linda Chena , Sunny Trana , Newman Chengb , Roman Wangb, Nikhil Singha , Taylor L. Pattic , Jayson Lynchd, Avi Shporera, Nakul Vermab, Eugene Wub , and Gilbert Strang - 30 may 2022.

[6]. Zhang, Dongxiang, Wang, Lei, Zhang, Luming, Dai, Bing Tian, and Shen, Heng Tao. The gap of semantic parsing: A survey on automatic math word problem solvers.