# Deep Neural Network based system for solving Arithmetic Word problems

by

Purvanshi Mehta, Pruthwik Mishra, Vinayak Sanjay Athavale, Manish Shrivastava, Dipti Misra Sharma

in

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
November 2017

# Deep Neural Network based system for solving Arithmetic Word problems

**Purvanshi Mehta** *
Thapar University

**Pruthwik Mishra** * and **Vinayak Athavale**
Language Technologies Research Center
IIIT HYDERABAD

**Manish Shrivastava** and **Dipti Misra Sharma**
Language Technologies Research Center
IIIT HYDERABAD

## Abstract

This paper presents DILTON, a system which solves simple arithmetic word problems. DILTON first predicts the operation that is to be performed ('-','+','*','/') through a deep neural network based model and then uses it to generate the answer. DILTON divides the question into two parts - worldstate and query as shown in Figure 1. The worldstate and the query are processed separately in two different networks and finally the networks are merged to predict the final operation. DILTON learns to predict operations with 88.81 % in a corpus of primary school questions. With simple similarity between the contexts of quantities appearing in the problem and the question text, we are able to identify 92.25 % of relevant quantities and solve 81% of the questions. Our code and data is publicly available.[1]

## 1 Introduction

In recent years there is a growing interest in understanding and generating natural language for the purpose of answering questions related to science and maths. Computers are better than humans in terms of both speed and accuracy at mathematical calculations but it is still a challenging task for computers to solve even elementary grade math-word-problems (Problems described in natural language). From the perspective of Natural Language Processing, mathematical word problems are challenging to solve as we need to reduce the natural language text to a set of equations which we can then automatically solve.
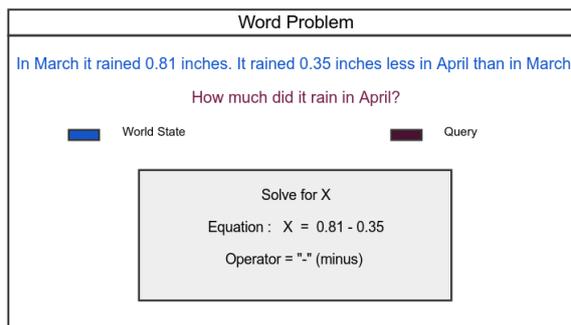


Figure 1: Example problem and answer generation

Arithmetic word problems can be solved with the help of the numbers mentioned in the text and their relationships through basic mathematical operations (addition, subtraction, division, multiplication). Arithmetic word problems begin by describing a partial world state, followed by simple updates and end with a quantitative question. For humans, understanding the language part is trivial, but the reasoning may be challenging; for computers, the opposite is true. Designing algorithms to automatically solve math and science problems is a long-standing AI challenge (Bobrow, 1964). Work done in this domain range from template-matching to narrative-building, integer linear programming and factorization. In symbolic approaches, math problem sentences are transformed by pattern matching or verb categorization. Equations are derived from the patterns. Statistical learning methods are employed in the paper (Hosseini et al., 2014) There has been work done in extracting units and rates of quantities (Roy et al., 2015); (Mitra and Baral, 2016) focus on addition-subtraction problems. We focus on solving problems with a single operation, (Koncel-Kedziorski et al., 2015) focus on single equation problems, and (Hosseini et al., 2014) focus on al-

---

gebra word problems.

Our system used GRUs and LSTMs to process the question and predicted the operation between the numbers mentioned in the text. Arithmetic word problems concisely describe a world state(WorldState) and pose questions(Query) about it. For example, Figure 1 shows one such problem. The described state can be modeled with a system of equations whose solution specifies the question's answer.

This paper studies the task of learning to automatically solve such problems given only the natural language with two operands in the question. The solution involves the understanding of the text. In our system, first of the question is divided into two parts WorldState(describes the quantities and how are they being modified) and Query(The quantity being asked). The WorldState and the Query are processed separately.

Our contributions are -

1. We present DILTON , a novel, fully automated system that learns to solve arithmetic word problems with two operators.

2. We used a Deep Neural Network based model to automatically predict the mathematical operation present in a arithmetic word problem.

3. We propose a simple and effective way of identifying relevant quantities in a word problem through similarity between context of each quantity and the corresponding question.

## Problem Description

We address the problem of automatically solving arithmetic word problems. The input to our system is the problem text P , which mentions 2 quantities num1, num2 . Our goal is to predict the operation between the two numerical quantities. Inputs to our model are in the form of a question which consists of a world state which describes the background of the question and a query which describes the quantity for which the question is being asked.

## 2 System Working

DILTON's working is shown in figure 3. Input as the math word problem is given and then the numerical quantities are separated from the text. Word problem is separated into query and the

world state. The world state is defined as the word problem without the final query which has information required to answer the query. We vectorized both the query, worldstate separately and then used our Deep Neural Network based model to predict the operation needed to answer the query. After predicting the operation the system applied it on the numerical operands to compute the answer to the problem.

## 3 Model

### 3.1 Architecture:

Out system is a pipeline consisting of three different modules that are detailed below.

### 3.2 Sequence Autoencoder

We used word2vec (Mikolov et al., 2013) to convert each word in the world state, query to its vector representation. We then used a sequence autoencoder (Dai and Le, 2015) with a GRU to encode both the world state and the query separately.

### 3.3 Combining the representations

We take the outputs of the sequence autoencoder for both the query, world state separately and combine them by doing an element wise sum.

### 3.4 Predicting the answer

We take the combined representation and then apply a GRU on it to get a vector representation for the combined (query,world state) The terminal layer in our architecture is a fully connected layer. It converts the output of GRU-RNN layer into softmax probabilities for each class.

### 3.5 Operand Prediction

In order to find the operands in a word problem, we need to first filter out irrelevant quantities. e.g John has 3 pens and 2 pencils. Jane have John 5 more pens. How many pens John have now? In this question, the quantity 2 is irrelevant which can be easily found out by a similarity match between the context of the quantity and the question asked. We experimented with different context window lengths across quantities and reported the results.

### 3.6 Training

We train this whole network end to end by using categorical cross entropy error and stochastic gradient descent. We use 30% Dropouts (Srivastava et al., 2014) for regularization and to prevent overfitting. We used 50 sized word2vec (Mikolov et al., 2013) embeddings and GRU's with 100 hidden nodes to encode both query, worldstate and trained the network for 40 epochs.
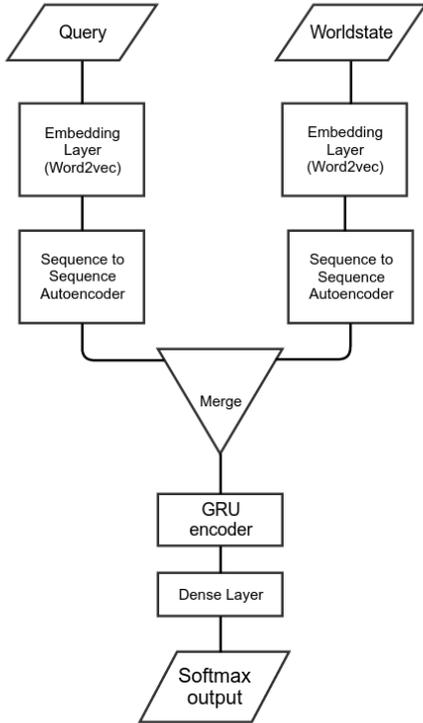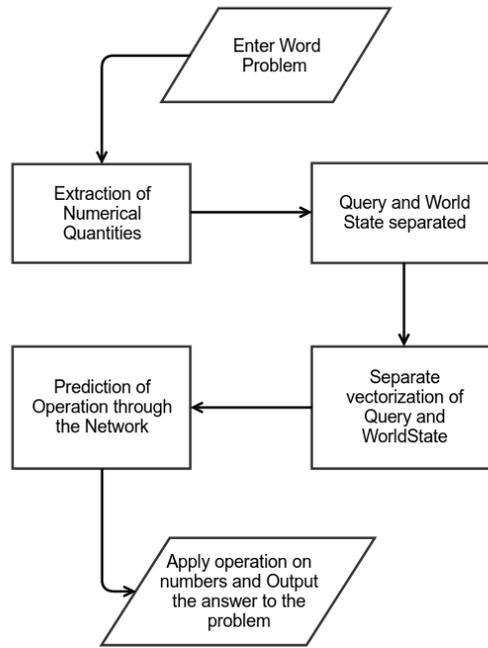
Figure 2: DILTON network



Figure 3: Workflow of DILTON

## 4 Experimental Study

In this section, we seek to validate our proposed modeling. We evaluate our systems performance based on the percentage of correct operator prediction. We do not directly evaluate our systems ability to map raw text segments into our representation, but instead evaluate this capability extrinsically, in the context of the aforementioned task, since good standardization is necessary to perform quantitative inference.

## 5 Experimental Setup Dataset

We have used the dataset provided by MAWPS (Koncel-Kedziorski et al., 2016). The dataset consists of the dataset included in singleop and addsub domain. The dataset consists of questions with two operands on which basic operations (addition, subtraction, multiplication, division) can be performed.These datasets have similar problem types, but have different characteristics. Problem types include combinations of additions, subtractions, one unknown equation, and U.S. money word problems. We randomly split the dataset into a dev set (for algorithm design, parameter tuning and debugging) and a test set. Our training set consists of 1314 questions and test set consists of 438 questions.

## 6 Baseline methods

We compare our approach with ARIS(Hosseini et al., 2014). The comparisons are mentioned in the table. Our system performs better than ARIS(Hosseini et al., 2014) in case when the question consists of two operands and a single operator. The neural networks performs better in case of learning the operations to be performed.

## 7 Evaluation Metrics

We get a training accuracy of 99.01%. and an accuracy of 88.81% on our testing data. We compare our results with ARIS which consisted of 395 questions and predicted the operations thorough verb categorization. ARIS dataset consists of 186 questions which our system cannot handle at the moment because three operators are present in the question. We compared our system against the 209 problems with single operation in the ARIS dataset.

## 8 Results

We evaluate DILTON in solving arithmetic problems in the dataset [2] provided by SingleOp

---

[2]We have not included the questions which had more than two numerical quantities

Table 1: Accuracies when trained on different models

| Model | Training | Testing |
|-------|----------|---------|
| GRU   | 99.54    | 88.81   |
| LSTM  | 98.33    | 87.90   |

and AddSub dataset . AddSub dataset was used by Aris(Hosseini et al., 2014) that achieved an accuracy of 81.2% for sentence categorization. DILTON shows significant improvement over their accuracies. It can learn to solve arithmetic word problems with an accuracy 88.81% on our testing data on a dataset consists of single basic operation.

Table 2: Comparison results

| System | Categorization accuracies(%) |
|--------|------------------------------|
| Aris   | 81.2                         |
| DILTON | 88.81                        |

We do not include the questions which consists of more than two numbers. We predict the final operator(addition, subtraction, multiplication, division) rather than categorizing every verb.

## 9  Error Analysis

DILTON encounters following errors while solving word problems:-

1. Questions which consist of more than two numerical values.

2. Question such as Raman had 2 chocolates and 4 apples. How many chocolates did Raman had? It cannot identify that there is no relation between 2 chocolates and 4 apples. Adding one more category of no relation can solve this.

## 10  Conclusion & Future Work

We propose a Deep learning based architecture on the task of math word problem solving. We di-

Table 3: Quantity Identification and Equation Formation Accuracy

| Context Window Length | Quantity | Equation |
|-----------------------|----------|----------|
| 1 | 92.25 | 81.92 |
| 2 | 79.35 | 70.47 |
| 3 | 77.74 | 69.04 |

vide the question such that the knowledge about the entities and the quantities asked are separated. Processing them separately makes sure that they don't share the same word embeddings. We show that deep learning models can significantly outperform many other approaches involving rule based systems or template matching or even traditional machine learning based approaches. As future work we will try to include the questions which can handle irrelevant information and questions with more than two numerical values.

## References

Daniel G Bobrow. 1964. Natural language input for a computer problem solving system.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, pages 3079–3087.

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 Dumas Ang. 2015. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, 3:585–597.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. Mawps: A math word problem repository. In *HLT-NAACL*, pages 1152–1157.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Arindam Mitra and Chitta Baral. 2016. Learning to use formulas to solve simple arithmetic problems. In *ACL (1)*.

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

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.