MathBot – A Deep Learning based Elementary School Math Word Problem Solver

{ Anish Kumar Nayak, Rajeev Patwari, Viswanathan Subramanian } {anishkn, rpatwari, vsub} @stanford.edu

Motivation

- Application of any learning algorithm to reduce natural language based math problems into equations is a topic of recent research
- Success of techniques such as Deep NLP, RNN flavors, Transformers etc. in this area to form a milestone towards general artificial intelligence
- Eventually build an end-to-end application, to assist elementary school parents and teachers

Datasets

- Source: MaWPS, Dolphin18k, Alg514, Draw
- Preprocessing done to extract, sanitize and number map dataset
- Each data point contains **input sequence (problem), output sequence (equation), and final solution**

WORD PROBLEM	Benny found 696 seashells and 109 starfish on the beach. He gave 248 of the seashells to Sally. How many seashells does Benny now have ?
OUTPUT EQUATION	$x = 696 + 109 - 248$
SOLVER OUTPUT	557

Dataset	Train	Dev
MaWPS-Full	2965	100
MaWPS-Elem	1811	100
Ext-Elem	2107	250
Ext-Elem-Mapped	2107	250
Combined	9568	1000
Combined-Mapped	9568	1000

Table 1: Composition of Datasets after Processing

References

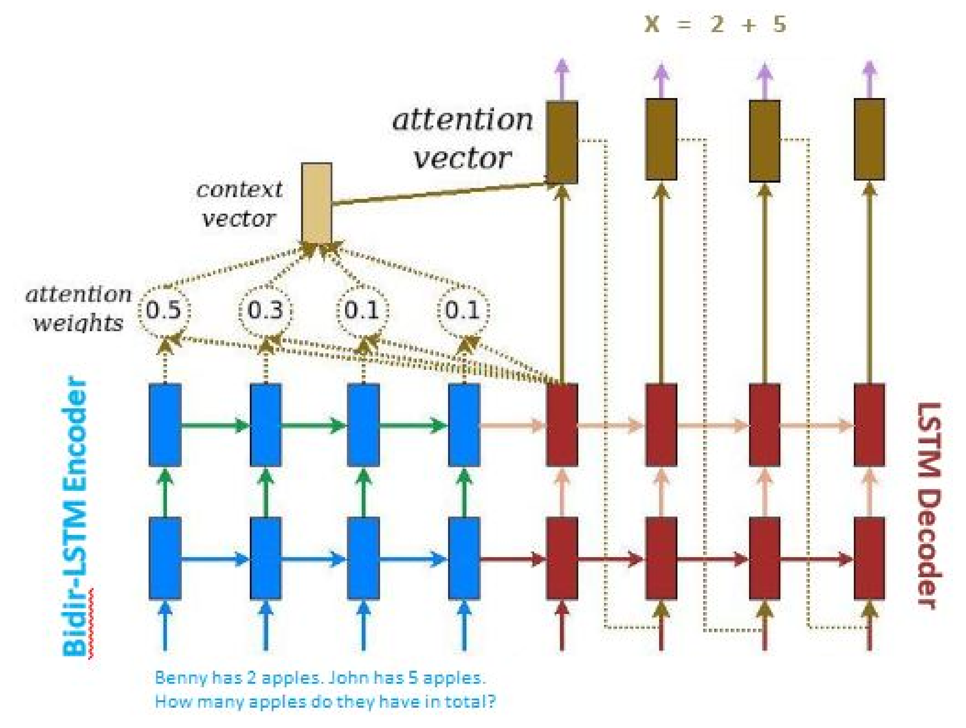
[1] Nicolas Chung Sizhu Cheng. Simple mathematical word problems solving with deep learning. 2019.

[2] Yan Wang, Xiaojiang Liu, and Shuming Shi. Deep neural solver for math word problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 845–854, 2017.

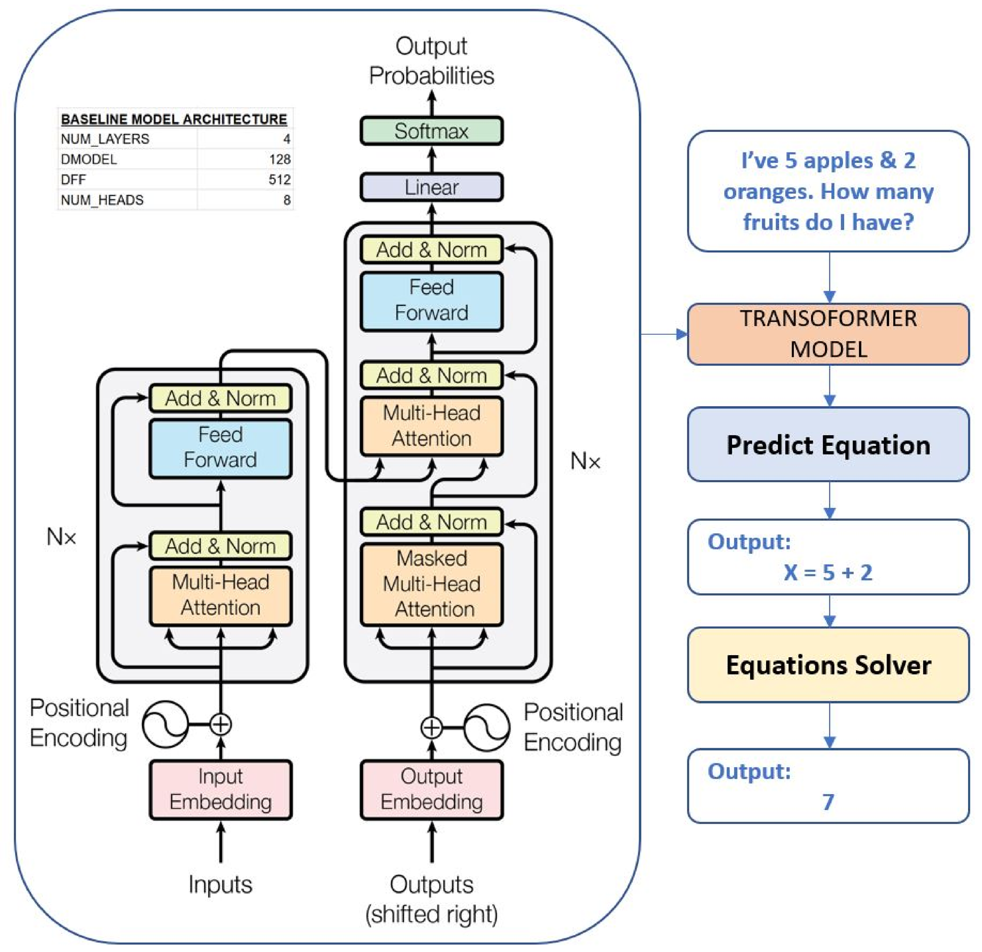
[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Methods

Bi-LSTM Encoder, LSTM Decoder Attention Model for Initial Setup



TRANSFORMER MODEL



Attention Function[3]

$$\text{Attention}(Q,K,V) = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O$$

$$\text{where, head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Loss Function & Accuracy Metrics

- SparseCategoricalCrossentropy
- SparseCategoricalAccuracy, Eqn Solver

Hyperparameters*	MaWPS-Full	MaWPS-Elem
32, 128, 0.5	34.25	27.38
32, 512, 0.5	86.86	89.81
32, 512, 0.3	88.38	88.92
32, 1024, 0.3	88.89	88.88
256, 256, 0.3	91.93	44.23

Table 2: BLEU scores with baseline Bi-LSTM, LSTM Attn. Model on MaWPS dataset (* *Embed Size, Hidden Size, Dropout*)

Dataset	BLEU-4	Solution Acc
Ext-Elem	64.73	57.82
Combined	16.89	9.85

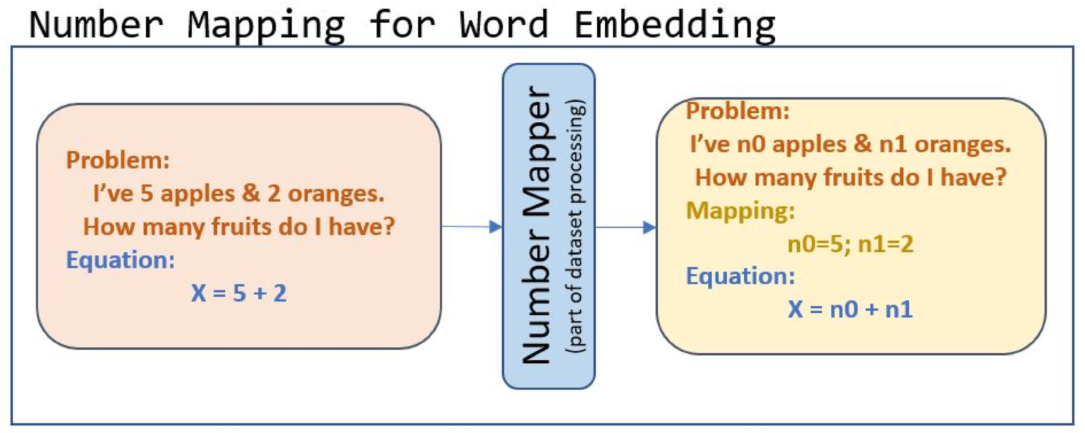
Table 3: Cumulative BLEU & Solution accuracy scores with baseline Transformer Model(*With batch size of* 64, *dropout of* 0.1, *and a custom learning rate schedule*)

HYPERPARAMETER TURNING AND IMPROVEMENTS

ARCHITECTURE & HYPERPARAMETER TUNING

Hyperparameter	Range of Variations (guided by error analysis at each step)
Number of Layers	4, 6, 8
Embed Size	64, 128, 256 512
Hidden Size	128, 256, 512, 1024, 2048
Number of Attention Heads	8, 16
Dropout	0.05, 0.1, 0.15, 0.2, 0.3, 0.5
Batch Size	32, 64, 128
Learning Rate	0.0001 through 0.1, and CustomSchedule $lrate = d_{model}^{-0.5} * min(step\_num^{-0.5}, step\_num * warmup\_steps^{-1.5})$

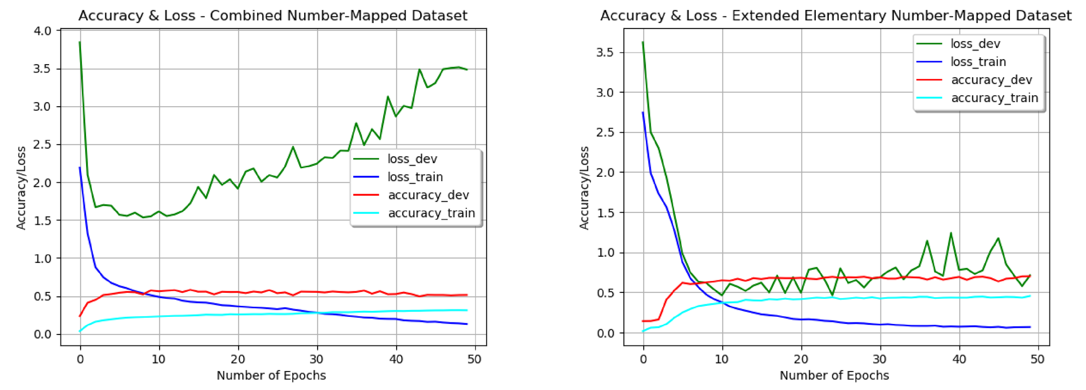
Number Mapping for Word Embedding



Future Research

- Implement beam search for transformer model to improve prediction quality
- Try larger AQUA-RAT dataset to extract equations, and obtain results
- Use transformer-XL and BERT with appropriate modifications
- Research on how to generalize to entirely new problem sets

Results



Plots using Tensorboard data generated for Model accuracy and loss on validation set

Dataset	Model Architecture	BLEU-4	Solution Accuracy
Ext-Elem-Mapped	0.1 - 6 - 256 - 1024 - 8	43.50	84.40
	0.1 - 4 - 256 - 1024 - 8	40.08	83.60
Ext-Elem	0.1 - 4 - 256 - 1024 - 8	27.66	63.20
	0.1 - 6 - 256 - 1024 - 8	25.94	58.00
Combined-Mapped	0.1 - 4 - 128 - 512 - 8	33.40	62.07
	0.1 - 4 - 128 - 256 - 8	31.60	61.45
Combined	0.3 - 6 - 256 - 1024 - 16	7.12	10.70
	0.1 - 6 - 256 - 1024 - 8	6.93	9.90

Table 4: Table of BLEU and Solution Accuracy scores
Arch: (dropout, layers, embed-size, hidden-size, num-heads)

Discussion & Conclusion

- Reproduced Bi-LSTM, LSTM Attn Model based work[1] for initial setup
- Analyzed BLEU scores and predicted equations to conclude that BLEU score alone is not sufficient evaluation metrics
- Developed our equation solver, and used it to compute solution accuracy scores
- Further error analysis on baseline transformer results helped us to develop number mapping technique
- Using number mapping for word embedding, we obtained much improved results
- Dataset with elementary problems in general gave better results since they were cleaner
- Combined dataset scores were lower, since several examples had inconsistencies in problems and equations especially in Dolphin18k dataset
- Tuned transformer with number mapping for word embedding, and eqn solver acc metrics resulted in improved prediction