Simple Mathematical Word Problems Solving with Deep Learning

Sizhu Cheng {scheng72} ,Nicolas Chung {sunchung}

Problem

Intelligence cannot be deprived of mathematical reasoning. People use their knowledge to solve extensive mathematical problem every day in real life. Mathematical problems are often stated in words in different scenarios, thus requiring problem solvers to extract information from the text and formulate in mathematical language to get the problem's answer. MWPs solving is believed to be challenging because of the semantic gap between the mathematical expressions and language logics [1]. We delved into simple algebra math problems and try to formulate mathematical equations that can solve the corresponding problem. We experimented with different deep learning model including bidirectional GRU seq2seq models and its variants, as well as Transformer.

Dataset

data set name	number of math problems	source
MaWPS	3914	MaWPS Repo
Dolphin 18K	18460	Yahoo
AQUA-RAT	100000	[1]

Data size in total: 45446

[1] MaWPS: from University of Washington

[2] Dolphin 1878: from community question answering site - Yahoo

[3] AQUA-RAT: Multiple Choices Question from [1]

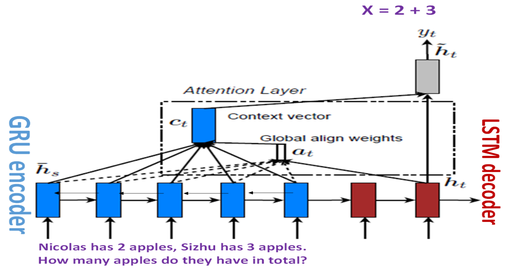
Preprocessing:

1. Use open source codes utilizing python libarary `urlib` to scrape down data from `answers.yahoo.com`
2. Regular expressions are used to extract all equations from 'rationale' of each piece of data in AQUA-RAT

Approach

Baseline:BiGLAtt

- Bidirectional GRU encoder,Unidirectional LSTM decoder, Multiplicative attention (BiGLAtt)
- embedding size: 32, Hidden size: 256
- Dropout : 0.5



Baseline model Architecture

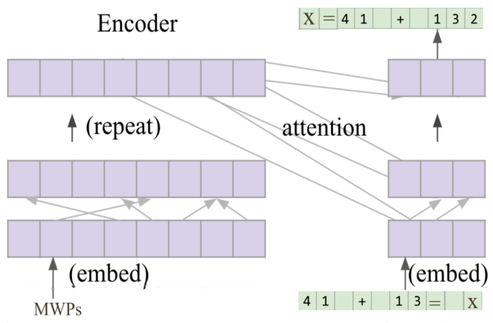
Approach

Baseline:BiLLAtt

- Bidirectional LSTM encoder,Unidirectional LSTM decoder, Multiplicative attention (BiLLAtt)
- embedding size: 32, Hidden size: 256
- Dropout : 0.5

Transformer

- Tensor2Tensor single gpu version
- Embedding trained
- Batch size: 2046, maximum length of the source: 256
- learning rate: 2.0



Transformer model Architecture

Quantitative Results

Models	Accuracy	Negative Log PerPlexity	BLEU
BiGLAtt	0.11	-1.0058479	28.14
BiLLAtt	0.16	-1.1486490	28.95
Transformer	0.6737	-1.7671001	N/A

Table 1: Models performance on dev sets

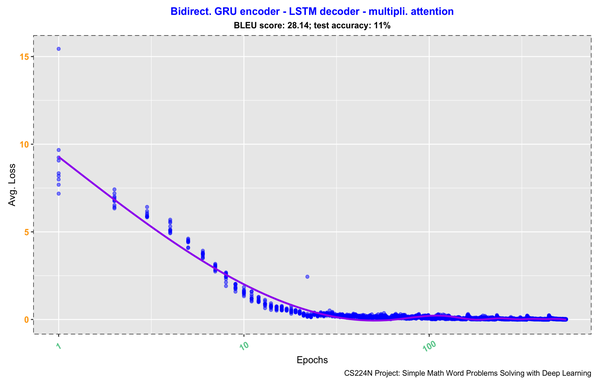
Analysis

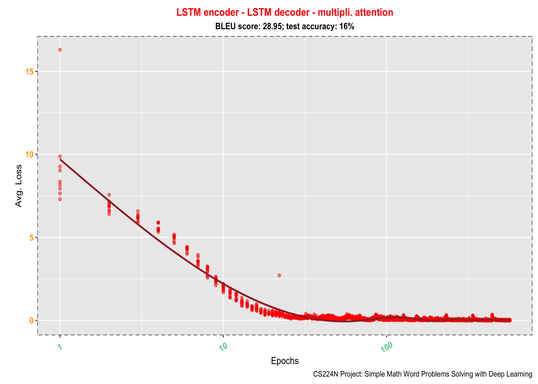
Q:Jason had Pokemon cards . He gave 9 to his friends. He now has 4 Pokemon cards . How many Pokemon cards did he have to start with?
Correct: X - 9 = 4
BiGLAtt Output: - 9 = 4 - X = 18
BiLLAtt Output: x - 9 = 4
Transformer Output: x - 9 = 4

Q: At Lindsey 's Vacation Wear , 0.375 the garments are bikinis and 0.25 are trunks . What fraction of the garments are either bikinis or trunks?

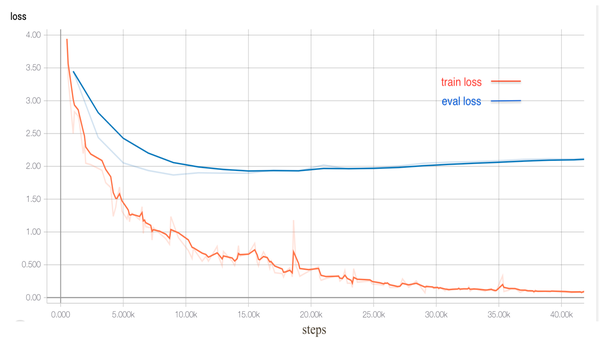
Models

BiGLAtt and BiLLAtt

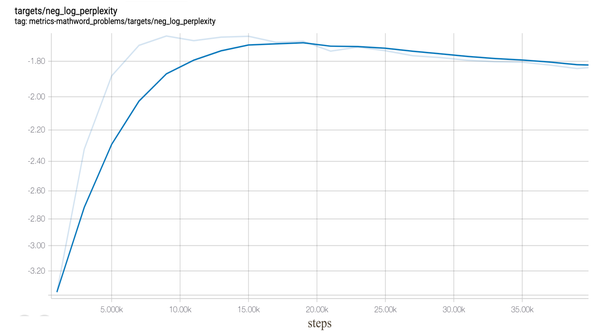




Transformer



Loss for both train and eval for transformers



Negative log perplexity for validation set during training

Early Stopping at 15,000 steps.

Analysis continued

Correct: X = 0.375 + 0.25
BiGLAtt Output: X = 0.25 + 0.25
BiLLAtt Output: X = 0.75 - 0.5
Transformer Output: X = 0.375 + 0.25

It seems like transformer can perform much better than the baseline when numerals appear are the decimals. Surprisingly, the results even works in the case:
Q: What is the fourth root of 400 over root 10?
Correct: x = 400 ^ (0.25)/sqrt(10)
Transformer Output: t = 400 ^ (1/4)/10 ^(1/2)
though the performance may be bad if solely evaluated using accuracy.
However, for problems with 'word numbers' and 'numerals' together: Q:If one third of 3/4 of a number is 21. Find the number?
Correct: 1/3 * 3/4 * x = 21
Transformer Output: x = 72
Equations are hard to be established and result is bad.

Conclusions

- Transformer improved performance a lot, though test accuracy is still not super satisfying
- Data is not perfectly correct because some targets are just 'equations' found in 'Rationale'. They may not be the actual equations to solve the problem, but just the brainstorming to the final answer
- Gap remained to fully capture the mathematical logics

References

[1] Wang Ling et al. "Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 158–167. DOI: 10.18653/v1/P17-1015. URL: http://aclweb.org/anthology/P17-1015.