JIT nudging for math word problems

Anant Mittal, Roshan Ramkeesoon, Sahil Verma

ABSTRACT

Given a math problem in English, we want to train a deep learning model to generate a string of steps consisting of mathematical operators to be used in order to solve the problem. At each step, the model should generate an operation (eg. add, divide) and its operands which can either consist of values from the word problem or results of previous operations.

The trained model can then be incorporated into the educational tools being used to teach word math problems in elementary schools. This tool will be used to provide hints to students when they get stuck at a particular step while solving the problem.

We pose this problem as NMT task and used several architectures to address it:

> We tokenized the input word math problem into words, and for the output, we tokenized them into both words and characters.

> We start with simplest word to character model

> Next we changed the tokenization of the steps to words, and hence trained a word to word model

> In the above models, we were learning the embeddings. We replaced learning with BERT pre-trained models. We trained two versions, with frozen and fine-tuning BERT.

> Finally we also attempted to solve this using Transformers

MATHQA DATASET

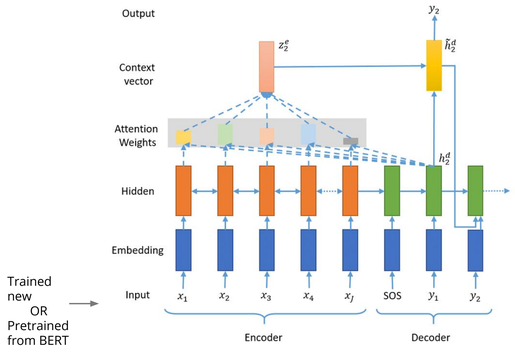
- 37k word problems with annotated formula that solves problem
- enhancement to existing AQuA dataset with annotated formulas

Example problem:
Two - thirds of a positive number and 16 / 216 of its reciprocal are equal. Find the positive number.

Linear formula:
`multiply(n0,const 3)|multiply(n1,const_2)|divide(#0,#1)|sqrt(#2)|`

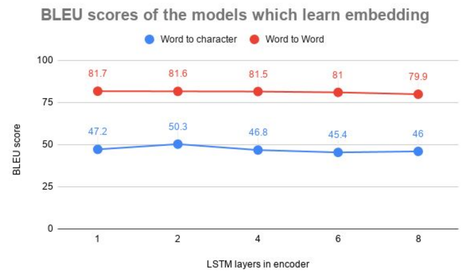
ARCHITECTURES

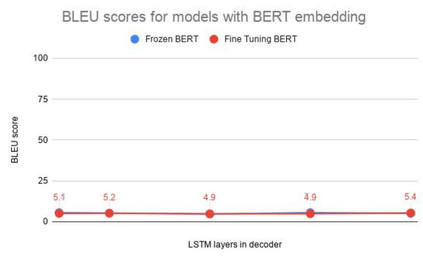
We used a variety of Seq2Seq architectures. In our encoder and decoder we kept the LSTM bidirectional and unidirectional respectively. We calculated BLEU-4 scores over the entire test set while modifying the number of LSTM layers in the encoder. As another modification, we replaced the embedding layer with BERT embeddings and trained our model with frozen as well as by fine-tuning the BERT layer.



Attention based seq2seq model for NMT
Source: Shi, Tian & Keneshloo, Yaser & Ramakrishnan, Naren & Reddy, Chandan. (2018). Neural Abstractive Text Summarization with Sequence-to-Sequence Models.

RESULTS





Sample Predictions

The following results are from the model which had the highest BLEU-4 score.

A shopkeeper sold an article offering a discount of 5% and earned a profit of 31.1%. What would have been the percentage of profit earned if no discount had been offered?

Ground truth:
`add(n1,const_100)|subtract(const_100,n0)|multiply(#0,const_100)|divide(#2,#1)|subtract(#3,const_100)|`

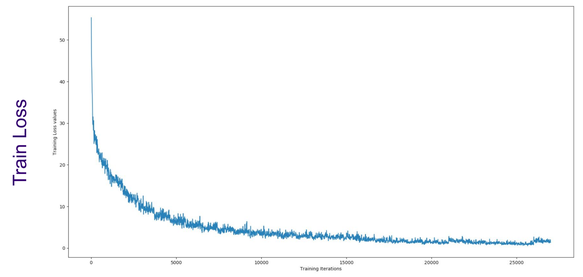
Prediction:
`add(n1,const_100)|subtract(const_100,n0)|multiply(#0,const_100)|divide(#2,#1)|subtract(#3,const_100)|`

What least number must be subtracted from 3832 so that the remaining number is divisible by 5 ?

Ground truth:
`divide(n0,n1)|floor(#0)|multiply(n1,#1)|subtract(n0,#2)|`

Prediction:
`divide(n0,n1)|floor(#0)|multiply(n1,#1)|subtract(n0,#2)|`

Train Loss Plot for Seq2Seq LSTM model with bidirectional single layer encode and unidirectional single layer decoder



CONCLUSIONS

- Attention based seq2seq model predicting formula as “words” without using BERT embeddings showed best performance of 81.7 BLEU score
- >1 LSTM encoder layers did not improve the BLEU score
- Predicting formula as characters showed worse BLEU score
- Using BERT embeddings for tokens in the encoder produced worse results than learning the embeddings
  - Math word problems is different context from corpus BERT was trained on
  - Fine tuning BERT did not improve results which probably is related to small dataset size (37k)

REFERENCES

[1] Amini, Aida, et al. "MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms." *arXiv preprint arXiv:1905.13319* (2019).

[2] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

[3] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

[4] Wolf, T., et al. "Huggingface’s transformers: State-of-the-art natural language processing." *ArXiv, abs* (1910).