Improved QANet on SQuAD 2.0

Default Project Track

Nicole Lee & Matthew Reed

PROBLEM

Question answering task:

- Models given a question and paragraph as input, then expected to extract the correct span from the passage that answers the question.

BACKGROUND

Initially, this question answering task was solved well for niche categories:

- Major League Baseball through BASEBALL
- Geological analysis of rocks from Apollo moon missions through LUNAR

Later, models like LSTM, BiDAF, and QANet emerged to perform well on a more broad SQuAD 1.0. Soon, the BERT model outperformed both LSTM and BiDAF. With the introduction of more unanswerable questions within the new SQuAD 2.0 dataset, these models began to show worse performance.

While some models have been tested on the SQuAD 2.0 dataset, the performance of QANet has not yet been researched. We propose that the introduction of the QANet model will improve performance for the SQuAD dataset.

METHODS

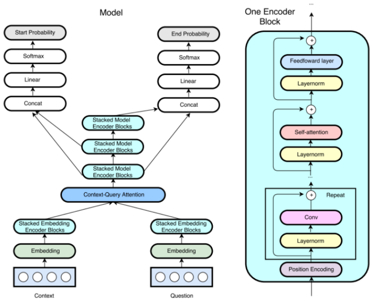

Figure 1. QANet Model Diagram [1]

EXPERIMENTS & ANALYSIS

We initially hypothesized that improving the BiDAF model by using a transformer encoder would beat the baseline. After hours of implementing, 3 experiments failed experiments, and office hours, we pivoted.

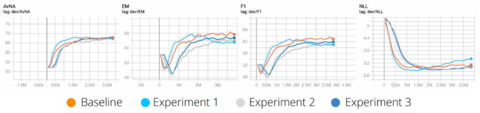

FIgure 2. Replacing BiDAF with Transformer

We implemented QANet from scratch instead by [1]:

- altering the embedding, embedding encoder, context-query attention, model encoder, and output layers.
- utilizing convolution and self-attention layers in the embedding and encoding layers as compared to RNNs for BiDAF.
- using built-in character-level embeddings:

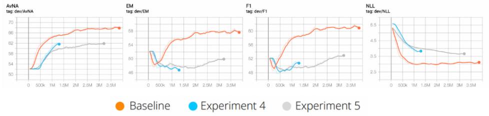

Figure 3. First phase of implementing QANet

Both experiments performed much worse than the baseline. We hypothesized that the poor performance was as follows:

- the sheer size of our model, as the 3 encoder blocks each contained 3 layers and the embedding encoder layer had 5 layers.
- lacking a linear projection layer to follow the embedding and BiDAF attention layers, exponentially increasing the embedding size

To improve our model, we then:

- simplified the model by decreasing the hidden size to avoid overfitting.
- implemented a linear projection layer to avoid creating exponentially large embeddings and number of parameters (Experiment 6).
- used a convolution layer as a projection layer (Experiment 7).
- utilized data augmentation for generalizability (Experiment 8).

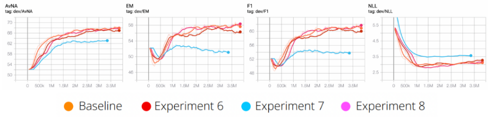

FIgure 4. QANet with simplification, convolution, and data augmentation

We altered the data augmentation implementation to also use learnable positional encodings to the encoding layer, as:

- the original architecture encodes using a sinusoidal lookup table.
- learnable parameters can better help with learning positional encodings.

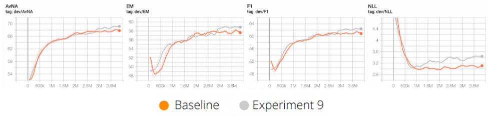

FIgure 5. QANet with data augmentation and learned positional encodings

With further analysis, we found that our final model:

- better answers "when," "which," and "who" questions compared to "why," as "why" questions commonly require more logical reasoning.
- struggles predicting answers to questions that do not explicitly begin with the 5W's or H. Most likely due to inability to generalize.

CONCLUSIONS

From our experiments and results, we found that:

- an improved BiDAF model with a transformer encoder does not perform better than the baseline.
- a QANet model with data augmentation and learnable positional encodings performs well on SQuAD 2.0, beating the baseline.

With more time, we would:

- test Transformer-XL techniques within QANet.
- change the loss function to better optimize F1 and EM scores.

[1] Yu, Adams Wei et al. Qanet: Combining local convolution with global self-attention for reading comprehension.1804.09541, 2018