QuAVONet: Answering Questions on SQuAD 2.0 with Neural Networks

Jesus Cervantes (cerjesus@stanford.edu))

PROBLEM DEFINITION

Objective: QuAVONet seeks to answer passage-based reading comprehension questions without the use of Pre-trained Contextual Embeddings (PCEs)..

Motivation: QANet initially worked fairly well on SQuAD 1.1, with few unanswerable questions. By combining QANet with Answer Verifier (AV) from U-Net, QuAVONet seeks to adapt QANet into the world of determining answerability.

DATA

The SQuAD 2.0 dataset is a set of 150,000 questions whose answers either lie as a span of the corresponding passage. New to Squad 2.0, 50,000 of these questions, can't be answered with the corresponding passage.

Answer Verifier

- **Component from U-Net [3]**
  a) **Use logistic regression to determine answerability**
- **Logistic Regression - x =** $(c_q; o_{m+1}; c_s; c_e)$ $c_q = \bar{\bar{S}}^T \cdot C$

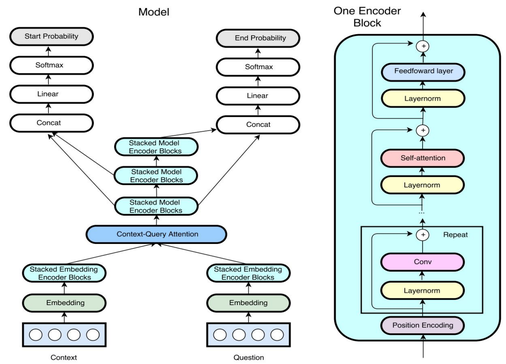
$$J(\theta) = \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)})) \quad c_s = p^1 \cdot B, c_e = p^2 \cdot B$$

- **Binary Cross-Entropy Loss**

$$L_{AV} = -(\sigma log(p^c) + (1 - \sigma)(1 - log(p^c)))$$

- **If P(answerable) < threshold, predict NA**
  a) **Threshold from paper = 0.3**

QANet



- **Encoder Block based on Transformer**
- **Loss function:**

$$L(\theta) = -\frac{1}{N} \sum_{i}^{N} \{log(p^1_{y^1_i}) + log(p^2_{y^2_i})$$

RESULTS

Various model Performance during training

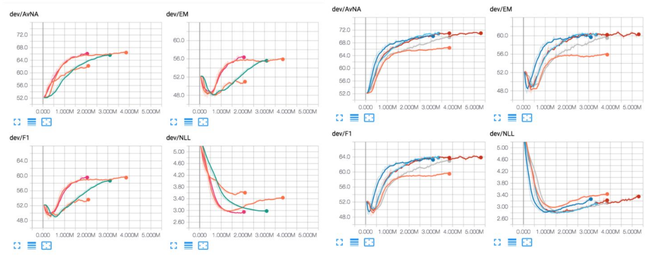


Figure 3: Plots of the AvNA, F1, EM, and NLL scores for the models, weaker ones on the left and stronger on the right. Left: baseline, orange; QARNNet, pink; AVQAEmb, mint; QAEmbEnc, orange (bottom) Right: baseline, orange; QuAVONet, gray then rust; QAEmb, dark blue; QANet 8-head light blue; QANet 4-head dark red

	Dev EM/F1	Test EM/F1
QANet (4 head)	**60.813/64.458**	**58.157**/61.39
QANet (8 head)	60.595/64.392	56.957/59.812
QuAVONet (4 head)	60.746/64.357	57.54/**61.40**
QAEmb Baseline	60.444/63.759	N/A

DISCUSSION

- 8-Head, d_model=128 QANet does not perform as well as 4-head QANet, d_model =96 QANet, likely due to slightly increased overfitting
- QANet significantly stunts training speed: 7 hours for 3M iters on QAEmb vs 25 hours for 4-head QANet on NV-6
- While adding QANet Embedding significantly boosts baseline performance (dark blue on right), adding the QA Embedding Encoder (bottom orange left) stunts performance
- Replacing the QA Embedding Encoder with RNN Encoder and convolution in QANet (pink left) does not boost speed or performance
- QuAVONet does not surpass QANet and takes more iterations to overfit and achieve comparable performance
- Adding AV to QAEmb (mint left)retains superior speed, but stunts performance

Conclusion

QANet vs QAEmb Baseline
While QANet and QuAVONet are both able to outperform the baseline, by supplementing the baseline with the embedding layer from QANet, it achieves nearly comparable results in a fraction of the time.

QANet vs QuAVONet
Interestingly enough, QuAVONet seems to be unable to surpass QANet, suggesting that the logistic regression Answer Verifier seems to hold back performance rather than improving it.

Future Work
My QANet runtime is much slower relative to RNNs than the original QANet, meaning there may be some underlying issue with my implementation impacting both speed and performance. Investigating this is a worthwhile future endeavor.

REFERENCES

[1] Minh-Thang Luong Rui Zhao Kai Chen Mohammad Norouzi Adams Wei Yu, David Dohan and Quoc V Le. Qanet: Combining local convolution with global self-attention for reading comprehension. 2018.

[2] Fu Sun, Linyang Li, Xipeng Qiu, and Yang Liu. U-net: Machine reading comprehension with unanswerable questions. 2018.