Integrating the QANet Structure into the Default Model BiDAF for Question Answering

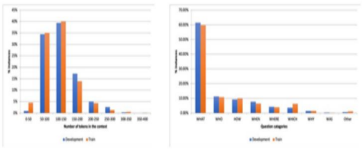
Stanford CS224N Default Project

Pham Thanh Huu
Department of Computer - Science Stanford University
huupt@stanford.edu

PROBLEM

This project addresses the problem that the default BiDAF machine comprehension and question answering model is RNN based and thus slow as it is not parallelizable. This project aims to integrate the QANet model, which uses convolutions and self-attentions to form a model architecture that is faster than recurrent-based approaches, into the default BiDAF model to enhance the baseline scores by tuning existing baseline model hyperparameters and changing the model architectures by adding new layers

DATA



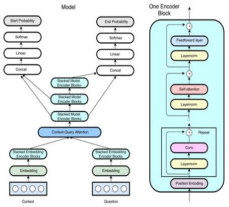
(a) Context lengths (b) Question categories

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.

DISCUSSION

- Used a robust baseline of the BiDAF model with word and character l evel embedding. While training batch size was reduced to 32, faced CUDA out of memory error for 64 batch size.
- Training was slow and took around 30 minutes to complete each epoch.
- Hyperparameter tuning in the original QANet model.
- Experimented with different encoder blocks in the model layer of QANet.
- Not much difference seen when using 5 and 7 encoder blocks.

QANet



$$p^1 = softmax(W_1([M_0; M_1])),$$

$$p^2 = softmax(W_2([M_0; M_2]))$$

$$p^1 = softmax(W_1([M_0; M_1])),$$

	Char-Embedding	Self-Attention	Batch size	Hidden Size	Optimizer	Learning Rate	# of heads	Dropout_prob
BIDAF (Starter code)	✗	✓	64	128	Adadelta	0.5	8	0.2
BIDAF (with Character Embedding)	✓	✓	32	128	Adam	0.001	8	0.1
QANet*	✓	✓	32	128	Adam	0.001	8	0.1
QANet**	✓	✓	32	128	Adam	0.001	8	0.1

RESULTS

Model	Dev(F1)	Dev(EM)	Test(F1)	Test(EM)	Evaluation
BiDAF (Starter code)	61.722	58.427	–	–	Baseline
BiDAF (with Character Embedding)	62.483	59.469	63.646	60.254	Success
QANet*	53.208	50.062	47.811	47.811	Failure
QANet**	56.119	53.487	54.708	51.784	Failure

Table 1: Overall Performance Comparison.

QANet*: QANet model with starter code along with character level embedding.
QANet**: QANet model with hyperparameter tuning discussed section 5.3

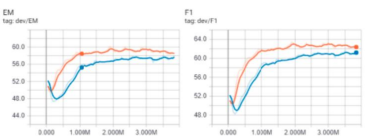


Figure 3: BiDAF with Character Embeddings. From the right, the graphs are produced on the dev set and represent EM, F1 and the loss. The orange curve represents BiDAF with word + character embeddings and the blue curve represents the baseline BiDAF

INTRODUCTION

In [1], Adams et al. proposed QANet, a question-answering model based on convolutions and self-attention instead of recurrent neural networks. At the time of publication of the paper (Apr 23, 2018), the most successful question-answering models were generally based onrecurrent neural networks (RNNs) with some attention mechanisms. One weakness of these models is that their recurrent nature prevent parallel computation, making training and inference slow. There were efforts to speed up the RNNs by avoiding bi-directional attention [2] or deleting the context-query attention module [3], but these models had to sacrifice accuracy: On the SQuAD v1.1 dataset, their F1 scores were around 77.

QANet solves this problem by moving away from RNNs instead of using convolutions and self- attention as the main building blocks. This feed-forward nature gives the model a significant speed advantage (reportedly between 3 to 13 times faster in training and 4 to 9 times in inference) over its RNN counterparts. Taking advantage of this speed boost, the authors trained the model with augmented data and achieved an F1 score of 84.61 on the SQuAD dataset, which was significantly better than the best-published result of 81.8 at the time.

QANet on PyTorch was implemented in this project and tested on SQuAD 2.0. While the transition to SQuAD 2.0 is straightforward, it is challenging to reproduce the performance, especially the speed, reported in the original paper.
A number of the author's peers in the CS224N class and open Github repositories reported the same issue.
While the issue was not resolved, a few factors which might Stanford CS224N Natural Language Processing with Deep Learning play an essential role in the model's performance were identified.

CONCLUSION

This paper walked through a custom implementation of QANet to speed up training and inference of reading comprehension models.
The core differentiator of the model is that it drops all recurrent layers in favor of convolutions and self-attention layers.
While the feed-forward nature of QANet is ideal for parallel computation, it is difficult to take advantage of this feature:
In contrast to the findings of the original paper, no speed increase could be observed on the baseline BiDAF model.
Some of the key findings is that character embeddings in conjunction with word embeddings help improve model performance regardless of the architecture.
It appears that in this implementation, since the model is capable of overfitting the data, more regularization might be needed in order to allow the loss on the validation set to continue to drop.
The use of multi-head attention gives a model a clear path to interpretability on examples, as it is easy to see which part of a context or query the model is focused on.
However, the model produced by this implementation does not match the performance of the original QANet. There could be potential changes to QANet in order to improve performance.

REFERENCES

[1] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension. CoRR, abs/1804.09541, 2018.
[2] Jonathan Raiman and John L. Miller. Globally normalized reader. In EMNLP, 2017.
[3] Dirk Weissenborn, Georg Wiese, and Laura Seiffe. Making neural qa as simple as possible but not simpler. In CoNLL, 2017.
[4] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi: "Bidirectional Attention Flow for Machine Comprehension", arXiv:1611.01603, 2016
[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser: "Attention Is All You Need", arXiv:1706.03762, 2017
[6] The annotated transformer. http://nlp.seas.harvard.edu/2018/04/03/attention.html .
[7] A pytorch implementation of qanet for machine reading comprehension. https://github.com/BangLiu/QANet-PyTorch.
[8] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable ques- tions for SQuAD. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics.
[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 5998–6008. Curran Associates, Inc., 2017. http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf
[10] A tensorflow implementation of qanet for machine reading comprehension. https://github.com/NLPLearn/QANet.
[11] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, arXiv:1607.06450, 2016