Comparing Approaches to Question-Answering on SQuAD 2.0

CS224N: Natural Language Processing with Deep Learning

Ray Iyer - `rri@stanford.edu`

Department of Computer Science, Stanford University

Question-Answering Task

Input: **question** and **context** (i.e., paragraph) of text

Output: A correct answer to the question, where the answer is a **span** (i.e., excerpt of text) from the context. In some cases, the question cannot be answered using the context.

Background

Recurrent Neural Networks (RNNs)

Traditionally, the most successful models for QA utilized a recurrent neural network to encode sequential input for downstream processing

Transformer

The Transformer has driven state-of-the-art improvements on adjacent tasks of language modeling, machine translation, etc.; here, we explore adapting its techniques of position encoding, feed-forward layers, and masked mult-head attention to the QA task.

Convolutional Neural Networks (CNNs)

CNNs are commonly used for visual analysis; along a sequence of words, they capture local textual structure.

Self-Attention

Self-attention learns the global dependencies between word pairs.

Data

Stanford Question-Answering Dataset 2.0

- **Size**: 129,941 train, 6078 dev, 5915 test examples
- **Example**: (context, question, answer) triple
- Three answers provided per example from different human labelers, to account for variance of reading comprehension and potential for multiple correct answers
- Train includes over 40,000 unanswerable questions

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] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. **Bidirectional attention flow for machine comprehension**. CoRR, abs/1611.01603, 2016.

[3] Pranav Rajpurkar, Robin Jia, and Percy Liang. **Know what you don't know: Unanswerable questions for SQuAD**. In Association for Computational Linguistics (ACL), 2018.

Neural Models

Bi-Directional Attention Flow (BiDAF) vs **QANet**

BiDAF

Input Layer
Concat[Proj(GLoVE Word Emb) + Conv2d(Char Emb)] -> Highway Network

Contextual Embedding Layer
Concat[Forward LSTM, Backward LSTM]
Model temporal interactions btw words

QANet

Input Layer
Conv1d[GLoVE Word Emb + Conv2d(Char Emb] -> Highway Network

Embedding Encoder Layer
Positional Encoding -> 4 x Conv1d -> Self Attention -> Feed Forward w/ residuals

Bi-Directional Attention Flow Layer
Context-to-query + query-to-context from similarity matrix

$\mathbf{S}_{tj} = \alpha(\mathbf{H}_{:t}, \mathbf{U}_{:j}) \in \mathbb{R}$ $\quad$ $\mathbf{G}_{:t} = \boldsymbol{\beta}(\mathbf{H}_{:t}, \tilde{\mathbf{U}}_{:t}, \tilde{\mathbf{H}}_{:t}) \in \mathbb{R}^{d_{\mathbf{G}}}$

BiDAF

Modeling Layer
Two-layer bi-directional LSTM; capture context word interactions conditioned on query

Output Layer

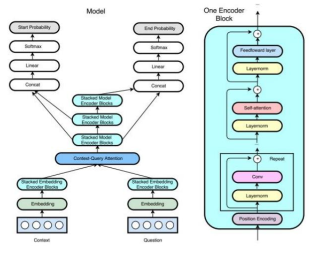
$\mathbf{p}^1 = \mathrm{softmax}(\mathbf{w}^\top_{(\mathbf{p}^1)}[\mathbf{G};\mathbf{M}]),$ $\quad$ $\mathbf{p}^2 = \mathrm{softmax}(\mathbf{w}^\top_{(\mathbf{p}^2)}[\mathbf{G};\mathbf{M}^2])$

QANet

Stacked Model Encoder Blocks
PosEnc -> 2 x Conv1d -> SelfAtt -> FF
7 blocks, applied 3x w/ shared weights

Output Layer

$p^1 = softmax(W_1[M_0; M_1]),\ \ p^2 = softmax(W_2[M_0; M_2])$

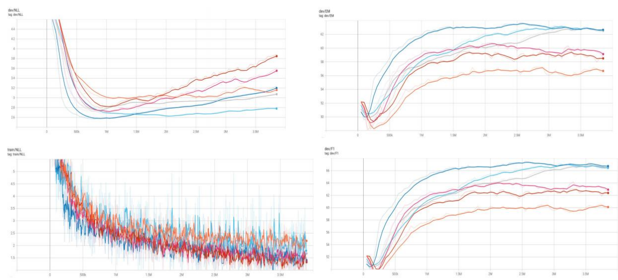


QANet Architecture (left) with Encoder Block in detail (right)

Experiments

Model	Batch Size	Train Time	Dev EM	Dev F1
Baseline BiDAF	64	3h11m	57.049	60.686
BiDAF + Character Emb	64	4h49m	59.368	62.839
BiDAF + Character Embedding + Fusion Fn	**64**	**4h12m**	**60.33**	**64.19**
QANet (2 heads, 3 model encoder blocks)	64	3h32m	62.95	67.00
QANet (4 heads, 5 model encoder blocks)	**32**	**6h42m**	**63.737**	**67.507**
QANet (8 heads, 7 model encoder blocks)	16	13h45m	63.27	67.13

- Each model was trained end-to-end with hyperparameters and optimizers specified by the original papers.
- *Gradient accumulation* was used to counteract the training instability introduced by smaller batch sizes for the large QANet models.



BiDAF (orange), BiDAF + char (red), BiDAF + char + fusion (pink), Small QANet (grey), Med QANet (dark blue), Orig QANet (light blue)

Analysis

- QANet generally performed higher than BiDAF, as expected.
- However, the incremental benefit of adding attention heads/encoder blocks was outweighed by the steep increase in training time for larger models.
- Adding a simple MLP fusion function to post-process the BiDAF attention output significantly increased performance over the baseline.

Conclusions

- The original paper's claim that QANet is faster to train than BiDAF is refuted in resource-constrained environments since batch size must be decreased.
- Larger model ≠ better perf; layer dropout could have improved dev results.
- The combination of convolutions, position encoding, and self-attention in QANet is promising as an alternative to traditional RNN encoders.