Coattention, QANet, and Data Augmentation for Question Answering

Wenqi Li[1], Scott Xu[1]

Department of Computer Science, Stanford University

Project Overview

Our project aims to explore the effect of different co-attention technique on the SQuAD question answering dataset, including the original Dynamic Coattention Network [1] and the QANet [2]. We also employ a different data augmentation technique from the one used in the QANet paper and is based on Easy Data Augmentation [3].
Our co-attention with dynamic decoder improved upon the baseline model by a small margin. Our final QANet implementation achieved high F1 score of 69 on the dev set and 68.71 on the test set.

Datasets & Metrics

We use the SQuAD v2.0 dataset for question answering and didn't include other question answering datasets or pretrained language model parameters. We use the word embedding and character embedding of the SQuAD dataset and the task is to optimize the F1 score of the answer predictions.

The baseline model is the provided BiDAF model, which uses a bi-directional LSTM as its encoder structure. It achieves an F1 score of around 60, and we consider any result above this an improvement.

References

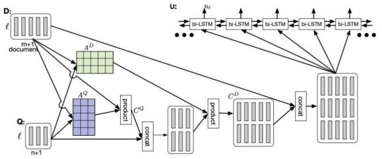
[1] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. In *International Conference on Learning Representations (ICLR)*, 2017.

[2] 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.

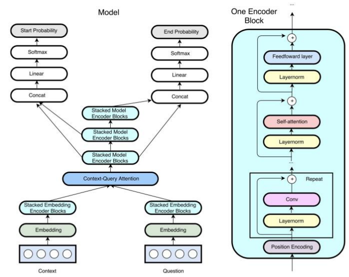
[3] Jason W. Wei and Kai Zou. EDA: easy data augmentation techniques for boosting performance on text classification tasks. *CoRR*, abs/1901.11196, 2019.

Model I: Coattention

We implemented the Coattention model with dynamic decoder. The model consists of two main parts: the Coattention-based encoder and the iterative dynamic decoder. The Coattention mechanism improves previous attention methods by proposing the concept of context-query attention in the QA task. The iterative dynamic decoder computes a start and an end score for each word in the context, iteratively update them, and finally predict the ones with highest scores as the start and end indices of the answer substring.



Model II: QANet



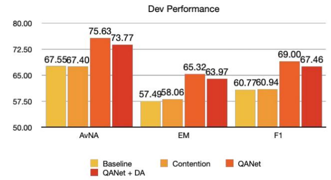
The QANet model follows the encoder-decoder structure and is made of five major components: the input embedding layer, the embedding encoder layer, context-query co-attention layer, the model encoder layer, and the output layer. For attention methods, the encoder layer uses self-attention techniques over the encoded representation of query and contexts, and the model encoding layer uses the co-attention techniques that is modified from the Dynamic Coattention Network. Our implementation uses 5 model encoder layers instead of 7 proposed in the original model due to memory constraints, and the model takes approximately 6 hours to run. We used a dropout rate of 0.1, batch size of 32 and model dimension of 128 in our best implementation.

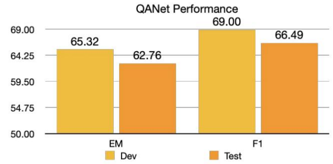
Data Augmentation

In addition to experimenting different attention-based model structure, we also augmented the training data to facilitate learning. Since our computational resource allows only implementation-wise easy and cost-efficient methods, we use a different data augmentation inspired by the Easy Data Augmentation paper [3].
In particular, we performed two forms of data augmentation:

1. Random swap: randomly take two words in the context and swap them
2. Random deletion: uniformly randomly delete words from each sentence in the context.

Results





Analysis

We notice that training on augmented data improves the baseline model but does not improve the QANet model. The reason maybe two-fold:

1. The augmented data is not significantly different or better than the original data: only slight modifications are done, which give the model limited new information to learn from.
2. QANet does not contain any RNN structure, so the random swap augmentation method may has little effect on it.

The performance could be possibly improved by ensembling QANet and RNet, or using a larger model dimension and number of encoding blocks in each layers in the QANet model.